Technical Note 264

HAYSTAQ

A MECHANIZED SYSTEM FOR SEARCHING CHEMICAL INFORMATION

ETHEL C. MARDEN

U. S. DEPARTMENT OF COMMERCE
NATIONAL BUREAU OF STANDARDS

THE NATIONAL BUREAU OF STANDARDS

The National Bureau of Standards is a principal focal point in the Federal Government for assuring maximum application of the physical and engineering sciences to the advancement of technology in industry and commerce. Its responsibilities include development and maintenance of the national standards of measurement, and the provisions of means for making measurements consistent with those standards; determination of physical constants and properties of materials; development of methods for testing materials, mechanisms, and structures, and making such tests as may be necessary, particularly for government agencies; cooperation in the establishment of standard practices for incorporation in codes and specifications; advisory service to government agencies on scientific and technical problems; invention and development of devices to serve special needs of the Government; assistance to industry, business, and consumers in the development and acceptance of commercial standards and simplified trade practice recommendations; administration of programs in cooperation with United States business groups and standards organizations for the development of international standards of practice; and maintenance of a clearinghouse for the collection and dissemination of scientific, technical, and engineering information. The scope of the Bureau's activities is suggested in the following listing of its four Institutes and their organizational units.

Institute for Basic Standards. Applied Mathematics. Electricity. Metrology. Mechanics. Heat. Atomic Physics. Physical Chemistry. Laboratory Astrophysics.* Radiation Physics. Radio Standards Laboratory:* Radio Standards Physics; Radio Standards Engineering. Office of Standard Reference Data.

Institute for Materials Research. Analytical Chemistry. Polymers. Metallurgy. Inorganic Materials. Reactor Radiations. Cryogenics.* Materials Evaluation Laboratory. Office of Standard Reference Materials.

Institute for Applied Technology. Building Research. Information Technology. Performance Test Development. Electronic Instrumentation. Textile and Apparel Technology Center. Technical Analysis. Office of Weights and Measures. Office of Engineering Standards. Office of Invention and Innovation. Office of Technical Resources. Clearinghouse for Federal Scientific and Technical Information.**

Central Radio Propagation Laboratory.* Ionospheric Telecommunications. Tropospheric Telecommunications. Space Environment Forecasting. Aeronomy.

---

* Located at Boulder, Colorado 80301

** Located at 5285 Port Royal Road, Springfield, Virginia 22151

NATIONAL BUREAU OF STANDARDS

Technical Note 264

ISSUED SEPTEMBER 27, 1965

HAYSTAQ

A MECHANIZED SYSTEM FOR SEARCHING CHEMICAL INFORMATION

Ethel C. Marden

Institute for Applied Technology
National Bureau of Standards
Washington, D.C.

For sale by the Superintendent of Documents, U.S. Government Printing Office
Washington, D.C., 20402 - Price 50 cents

Contents

HAYSTAQ
A MECHANIZED SYSTEM FOR SEARCHING CHEMICAL INFORMATION

Ethel C. Marden

ABSTRACT

HAYSTAQ is a comprehensive computer system for searching chemical information and is particularly directed toward the stringent requirements of the U. S. Patent Office. The greatest activity to date has been in the design of a satisfactory method to search for chemical structures. A structure diagram is considered as a network, where the atoms or functional groups are the nodes and the bonds between them the links. The search algorithm consists of attempting to match, via a topological tracing, a question network (structure) against each structure in the file of chemical compounds stored on magnetic tape. The structure search includes provision for Markush structures and other generic concepts. Each of 385 questions was matched against a file of 2,400 entries containing (because of the Markush feature) effectively 162,000 compounds. The continuation of this work includes the use of the Hayward linear notation as input and extension of the search routines to other kinds of information associated with chemical structures.

I. INTRODUCTION

HAYSTAQ was, and is, intended to be a comprehensive system for searching chemi-
al information. In particular, it is directed toward the peculiarly stringent require-
ents of patent searching. In recognition of the large volume of material to be searched
order to determine whether or not a patent should be issued, investigation has been
ade of mechanized means for searching all, or a large portion of, the material in the
otential) file or library of information available to the searcher.

Three people were chiefly responsible for the initial system design of HAYSTAQ:
rs. Ethel C. Marden, National Bureau of Standards, and Messrs. Herbert R. Koller, and
arold Pfeffer, U.S. Patent Office. Others in both organizations have been concerned with
riting machine programs for the various data preparation routines, with analysis of
. S. patents to obtain pertinent information for the file, and with implementing later
ages of the evolving system. Their names are listed in Appendix A, and their work is
eferred to in Sections VII and X.

HAYSTAQ's history has been one of evolution, and as such the nature of the project
as reflected its changing emphases, although the basic objective of the research has
emained constant: to develop a comprehensive system for searching chemical informa-
on. Investigations in connection with the early work described in Section IV made possi-
le the development of the chemical structure search program, described in Section VI.
he results of that work and the insights gained thereby, described in Section IX, pointed
the direction which the project is now taking, as discussed briefly in Section X. As in
he various stages in the life cycle of Lepidoptera, where an egg changes to a caterpillar,
hence to a cocoon, and finally to a butterfly, so does the character and form of this pro-
ect change as it progresses from one stage to the next, in the attempt to evolve

a complete system -- a system which will be of assistance to the patent examiner in making searches of chemical information.

A description of two stages of HAYSTAQ up to the present time forms the basis of the major portion of this report; the third stage (the present) is now under development, and is referred to in both Sections IX and X.

II. BACKGROUND

The U. S. Patent Office and the National Bureau of Standards have been investigating for the past seven or eight years the feasibility of mechanizing some portion of the literature searching and information retrieval operations of the U. S. Patent Office. Recommendations for such a collaborative activity were made to the Secretary of Commerce by the Bush Committee as long ago as 1954 [1]. The U. S. Patent Office has experienced increasing difficulty in making adequate searches of the technical literature, whose volume increases at a constantly accelerating rate.

The requirements of patent searching are more stringent than those of the majority of other types of literature searches. There is a statutory requirement that patentability be predicated on novelty, utility, and inventiveness. If a patent examiner determines, after search, that a concept is novel, he is still constrained to find the nearest similar related concepts previously known, patented, or published. Theoretically, when a patent is issued, it is assumed that there is no precedent concept disclosed anywhere in the literature.

It is probable today, as it has been true in the past, [1] that the examiner, in coming to his decision, spends better than 50 percent of his time in searching for possibly relevant references. Today, aided by the Patent Office classification scheme, only a very small fraction of search time is devoted to locating the subclasses in which the first hundred or so of the most likely references may be found. The problem in assisting the examiner is therefore not only to sharpen the identification of the most likely, but to narrow the search to the 5 to 15 truly relevant.

The U. S. Patent Office has now issued more than 3,000,000 patents. In addition, its library maintains files of foreign patents, numbering more than 6,000,000, as well as many thousands of technical journals and books. In making patent searches, the examiners must not only search foreign and domestic patents, but must turn for knowledge to periodicals, textbooks, catalogues, abstracting services, and other forms of publications. All areas of technology are represented in the library which contains information on such diverse subjects as small manufactured articles, dress designs, new drugs, electronic computers, and even submarines. Using their own library as the principal store of information, U. S. Patent Office examiners make about 2,000 searches each day, each one of which may require finding answers to from one to more than 20 different questions (claims). Although chemical patents account for only 12 percent of the previously issued U. S. patents, 25 percent of the current searches are made in the field of chemistry.

These searches may vary from a very simple identification of a specific situation to a very complex search consisting of a generic disclosure of a series of related processes, where several examples may be given to illustrate each such process. The generic aspect may take the form of a genus in the usual sense, or it may be expressed in the form of a Markush[1/] type of structural formula, where a number of compounds declared to be equivalent may be disclosed in a single structural formula or diagram. The type of search which must be made, then, may vary from the most specific viewpoint to the most

1/ See page 20 for a more detailed definition of "Markush"

generic, and may involve general combinations, subcombinations, equivalence between concepts, negative concepts, contrived or synthetic arrangements, as in the case of a Markush group, and various combinations of the above.

Some preliminary investigations of the problems of patent searching, and of mechanizing portions of the searching of chemical information, were initiated by the newly formed Office of Research and Development, in the U. S. Patent Office, as long ago as 1956 and 1957 [2, 7] . .The first attempts at mechanization of document searching by HAYSTAQ consisted of a serial inspection of each document in the file. It is recognized that the most efficient procedures for human operation are not likely to coincide with the most efficient ones for machines; later studies are attempting to take greater advantage of machine capabilities to improve on the serial inspection practices. The importance of mechanizing at least some portion of the present search process should be obvious. Any decrease in the time of the human examiner's search will effectively expand the resources of the examining staff and so decreases backlogs, thus speeding the issuance of patents so vital to industry in an expanding economy.

III. APPROACH TO HAYSTAQ

The stated purpose of HAYSTAQ is to develop a comprehensive system for searching chemical information; it is expected that some of the principles embodied in this development will find useful application in other subject matter areas.

The subject matter of the first stage of HAYSTAQ was intended to cover all document content within the field of chemically pertinent information. An initial attempt was made to represent the nature of all of the different kinds of chemical information contained in a document by means of a list of descriptors. Such terms were of necessity very general; it is noted that only eight types of descriptors were provided to describe the spectrum of information contained in technical documents (see page 5). Although a computer program was written for the first stage and "debugged" against test data accumulated from actual patents, the experience with the test data seemed to indicate that the actual program would have little utility for the patent examiner, i.e., it would not necessarily decrease his present burden of having to examine personally and in detail several hundred documents. A more useful approach to the problem of searching chemical information might result from concentration on a detailed mechanized search of chemical structures.

It was recognized that there are many different kinds of elements of information which are of importance to the examiner in making searches in connection with applications for new patents. Some such elements are easy to define and are well-structured; some examples of this kind of information are those which can be expressed in terms of numbers, e.g., physical constants and empirical formulae, and those which can be represented by unambiguous sketches, e.g., a chemical structure diagram. It is easier to devise a unique digital representation for these kinds of well-structured information, albeit this sometimes becomes a contrived and lengthy procedure, than it is to so represent the amorphous, or unstructured, information which is contained in, say, natural language text. 1/ This is particularly true in those instances where patent lawyers disagree as to the interpretation of the semantic connotations.

1/ For instance, in the structural formula for a chemical compound, a sulphur atom unequivocally is or is not present; but what can be objectively determined from a natural language expression such as "possible side effects may include mild emotional disturbances"?

With these considerations in mind, it was decided to embark on a concentrated attack on the problems of chemical structure searching of organic compounds by mechanized means. This decision was both fortuitous and expedient for several reasons: (1) chemistry represents the area of greatest single activity in the U. S. Patent Office, with more than 25 percent of present grants of patents in that area, and therefore, progress in that area could be of great assistance to the U. S. Patent Office; (2) chemical structures are the heart of the problem in chemical information searching, and the requirement for structure searching cannot be ignored in any large-scale attack on the problem; 1/ (3) chemical structures can usually be represented uniquely, completely, and unambiguously, and thus can be transformed into a digital representation for machine manipulation; (4) structure searching may be said to concern itself with the common medium of expression among chemists, namely, structural formulas, and structural formulas represent a kind of information which lends itself relatively easily to the development of a systematic set of rules for computer manipulation; and, finally, (5) a systematic machine-usable representation of structural formulas offers an effective means of entry into a file which consists of structures keyed to other kinds of information. (This important implication will be discussed in more detail in later sections of the report.)

Accordingly, the initial objective of stage two of HAYSTAQ was to develop a mechanized chemical structure search program for organic compounds 2/; a part of that objective was inclusion in the program of several features which some of the U. S. patent examiners believed to be peculiarly suitable to their requirements. The principal such features were the ability to search for Markush structures and the ability to make generic searches. 3/

All of the computer programs for stages one and two of HAYSTAQ have been written for execution by SEAC, an electronic computer at the National Bureau of Standards. SEAC has now been retired, and present work, referred to in Section X, is being programmed for the PILOT research facility, also at the National Bureau of Standards. Both machines permit the addition of experimental equipment as dictated by the requirements of particular programs.

IV. EARLY HISTORY OF HAYSTAQ: STAGE ONE

The first computer program to be written for HAYSTAQ was a search routine for matching a list of descriptive terms [3]. There were planned for the system at that time four computer programs: two were for preparation of disclosure 4/ and question data, respectively, the third was the search routine itself, and the fourth routine, called the "Checkout Routine", was to be written for the purpose of validating apparent answers to questions which had been discovered by the search routine. Only the search routine was actually written for the computer, and machine runs of the program were made against test data for program debugging only. Certain features of the search program are worthy of mention here because they were good enough to build on and so would have utility for future work.

1/ Because it is desirable to concentrate on the problem of reducing the present search load.

2/ Their structures are usually well defined and are generally known.

3/ These constitute important differences between HAYSTAQ and any other approaches that have yet been made in mechanized chemical information searching.

4/ A "disclosure" is any item of information contained in a document; as used here, a disclosure is any encoded item treated as an identifiable unit in data preparation, search, and subsequent retrieval.

Encoded contents from patents or other documents were arranged serially in the disclosure file of information to be searched by the computer program. There were several levels of organization of the information contained in any such entry. The largest segment of information in the document treated as a unit was a process, including all of the disclosed steps. Process information was not treated in depth in this primitive system [4]. The presence of process information, of botanical information, or any other category of information indicated in the list below, except for 2--Empirical Formulae and 3-- Chemical Information, was used only for the purpose of indicating the presence of that category of information in the document. Thus, they were of value in "screening", or rejecting from consideration documents which contained no listing of that category of information.

The process might be considered as a recipe for a multi-stage development, where at each stage there was one or more compositions. The composition then was at the second level of the information contained in the document and represented physical admixtures of materials. Each process was thus divided into compositions (which change from stage to stage), and each composition was further subdivided into ingredients, or items. The detailed information describing each such item consisted of a series of descriptor words which fell into eight different categories. As many of the selected terms of each category were employed as was required to furnish a description of the encoded item adequate for preliminary tests. The categories were as follows:

1 -- Index Number (a unique identification number)

2 -- Empirical Formula

3 -- Chemical Information

4 -- Botanical Information

5 -- Zoological Information

6 -- Anatomical Information

7 -- Process Information

8 -- Miscellaneous

Only the information contained under descriptor numbers 2 and 3 was treated in detail; when either of these numbers was recognized, the program entered subroutines for additional detailed matching of empirical formula and/or more specific "chemical descriptor" information, respectively, as called for by the question.

Questions to be posed against the file were manually formatted in the same manner as the disclosure entries which had been accumulated to form a file. The search progressed by means of a step-by-step serial matching operation at each level [1/] of the file organization. In the computer operations, all apparent "hits" at the item level were recorded.

It was intended that the checkout routine then take as input data the tables of tentative matches between question items and disclosure items. The purpose was to ascertain whether there was a sufficient total number of matching items when, for example, one question item was matched by several disclosure items or, especially, when several question items which required individual matches found a match in the same single disclosure item. However, the actual machine instructions for the checkout routine and for the question and disclosure data preparation routines were never written for the following reasons: On the basis of the examining experience of the two research chemists from the

1/ If a process was present, the matching of the compositions contained in it took place. The matching of each composition was carried out by an item-by-item match.

U. S. Patent Office and on a further examination of the full range of data accumulated for the test files, it appeared that because of the breadth of approach embodied in the search routine and in the data employed, the usefulness of the first program would not measure up to the real needs of the patent examiner.

Although the primary purpose both of the preliminary investigations and the subsequent ones has been to work toward a mechanized system which will be functionally adequate and thereby be of assistance to the patent examiners, the investigators at the same time have not been unmindful of questions of economic feasibility -- a consideration which must prevail in any operating system. Therefore, subsidiary objectives have been concerned with the design and test of techniques which will (1) by-pass searches if it can be determined in advance that such searches will not provide answers to the question, (2) terminate as quickly as possible those searches which will not find an answer, and (3) introduce if possible a random - access selection, or direct "fingering", of file items, where possible, in order to avoid the serial inspection of all data contained in the file.

Therefore, in the first computer program of HAYSTAQ, information was sorted and ordered at all levels for the purpose of terminating more quickly those searches which could not provide answers; ordering of the data permitted the search to skip to the next item when non-relevant material was reached in the list of data. Screening was also carried out on various levels, again for the purpose of terminating as quickly as possible those searches which could not produce answers to the question being asked.

A patent examiner is of necessity interested in the relationships among items in the file; some such examples are alternative relationships. There may be several items in an "and" relationship where one or more of them may be in an "or" relationship with each other; e.g., A and B and C and D and either E or F; again, A and B and C or D or E. Provision was made for including such associations in the file and in the question, and for searching on the basis of such relationships. Provision was also made for the ability to search for a "teaching" 1/ of equivalence in a disclosure document, that is to say, that the disclosure document itself provides the statement that one compound may be substituted for another in a given set of circumstances. At the same time, a questioner might ask for a negative teaching; that is, any document which would provide an acceptable response to the question must contain the statement that a certain thing is not permitted to be present under given circumstances. Another feature provided by the program was the facility to ask that a specified thing be absent; in this case, the questioner could ask either that the document answering the question "teach" that the specified thing be absent, or else that it simply fail to mention it. In some situations, the absence of particular ingredients is as important to the searcher as their presence in other cases.

The negative and absent characteristics were included as an integral part of each question descriptor, and for each disclosure descriptor there was a negative or positive indication. A formalism for handling the 15 different combinations of these relationships for searching purposes was developed for inclusion in the search program. The five question characteristics and three disclosure ones are listed below.

1/ A "teaching" is an affirmation of a positive statement made in a disclosure document.

Question Item	Disclosure Item
All descriptors positive	All descriptors positive
Some descriptors positive, others negative	Some descriptors positive, others negative
All descriptors negative	All descriptors negative
Some descriptors positive, others absent	
All descriptors absent	

Each of the 15 different combinations of question and disclosure characteristics required a slightly different search procedure, and the computer program provided for all combinations. When the characteristics of question and disclosure were determined, the required search pattern was selected by opening the gate to that path in the program for the particular variation desired, with the resultant closing of the other 14 gates.

As noted above, no large-scale testing of the computer programs for stage one of HAYSTAQ was attempted. On the basis of the computer runs made with the test data, it appeared that their discriminating power would not be great enough to give much assistance to the patent examiner. As an interim goal, it was decided that it was necessary to concentrate on those search problems that would be involved if category 3, namely, that of chemical structure, would be required to answer many specific questions. However, the experience so gained in operation with the test data was valuable for three principal reasons:

1. The examination of recent patents for possible entry into the disclosure file revealed shortcomings in the system before a commitment of funds or manpower to implement it had been made.
2. The experimental operation provided considerable insight into many of the difficulties surrounding the problems of providing the working examiner with a useful mechanized tool.
3. Even the limited experience with use of the test data revealed the power of the simple screening techniques employed for minimizing search times.

The investigators turned their attention to the question of how to capitalize on the experience gained with stage one of HAYSTAQ in order to move toward a system which might be used by the working examiner. The rationale for that approach and the description of the system which evolved is the subject of the next Section. Although the next phase of HAYSTAQ employed a different approach from the recipe-type model, it is expected that the philosophy of the earlier approach will be an innate part of future work with processes and reactions. Indeed, it subsumes the chemical structure of stage two. The structure search is concerned with ingredients of known structure: With the structure as the point for file entry, future work will concentrate on moving outward to include larger mixtures, until the search can again encompass the process level; the distinction between that and the very early work will then be that the later work will have specificity at every level.

V. SYSTEM DESCRIPTION OF STAGE TWO OF HAYSTAQ

Progress made on stage two of HAYSTAQ was reported in a paper presented at the Third Annual Meeting of an International Committee of Patent Office Experts Concerned with the Promotion of Cooperative Research Programs in Information Retrieval (ICIREPAT) in September 1963 [5]. One of the stated purposes of HAYSTAQ is to simulate the search which a patent examiner now makes manually. It is not intended that mechanized programs duplicate the human's thinking processes, but rather that the machine be able to locate all pertinent references by adequately efficient mechanized procedures. Thus, it is desirable to develop a system which will permit the examiner an acceptable degree of flexibility in putting questions to the file. Constraints on the manner of using mechanized files can only promote reluctance on the part of the examiners to employ mechanization.

The same file entry may answer a large number of questions, each reflecting a different interest on the part of the examiners formulating the questions. There are intimate associations between (1) the needs and the habits of use of examiners, (2) the manner in which they question the file, (3) the nature of the file organization, which must reflect use habits, (4) the character of the search system, which must interact with each of the first three, and (5) the subject matter being explored. The ingenuity constantly exercised in phrasing patent claims and the constantly shifting focus of interest in industry result in the continuous generation of new ways of expressing essentially the same or related ideas [6, 7]. It is therefore important to make the file contents as invariant as possible, consistent with the shifting patterns of language habits of examiners. At the same time it is desired to give wide latitude to the questioner, even though this flexibility may require added ingenuity on the part of the questioner to elicit desired items from the file. This approach requires that question formats contain more flexibility than the disclosure formats making up the file of information to be searched.

It was desired to enter into the file information which could be arranged in a systematic way and which could be recovered by means of a comprehensive set of rules. Any search scheme which was devised had to recover all possible answers to questions from the entries existing in the file. It was decided to enter into the file at this time information which lent itself to preciseness of expression so that there was little likelihood of ambiguity which could create either of the following situations:

1. Spurious responses (false drops) to a question or, worse,
2. Failure to find a legitimate answer which existed in the file.

The immediate goal of stage two of HAYSTAQ was to devise a satisfactory procedure for searching chemical structures, since this course of action would satisfy the two criteria listed above. This would also have the additional advantage of extracting from the entire problem a manageable-sized portion for an initial trial of mechanized chemical information searching. Other considerations were mentioned in Section III.

To supplement the structure search program, a series of computer programs was created. Following is a list of the various programs making up the HAYSTAQ system in stage two, with a brief description of what each accomplishes. There is a much fuller description of some of the routines in Section VII.

1. Structure search program. This program is defined in detail in Section VI. It is a detailed chemical structure search program which was carried out on the SEAC. It considers a chemical structure diagram as a network, where nodes of the network are functional groups, as defined in the detailed description of the structure search routine in Section VI. The bond connections between the functional groups may be considered as

the links between the nodes of the network. To match like structures, an algorithm was developed to execute a topological tracing of one network against another. The structure search program assumes it is working with error-free formatted data; the procedures for assuring that such a file is available are built into the machine programs described next.

2. SWEEP. SWEEP is an error detection routine which scrutinizes chemical structure descriptive information prepared by a chemist after he has analyzed the original document content and selected from the document all implicit and explicit references to organic chemical compounds. The information is then encoded in the manner described in Section VII to provide a complete description of the compound in a form which can be read by the computer. SWEEP calls in the data describing each such compound in turn, and subjects the data to a detailed examination for possible errors. SWEEP is able to identify a total of 88 different kinds of errors existing in the manually prepared data defining the structure; SWEEP also assists the analyst in pin-pointing the location of erroneous information by printing out the computer words containing such information. The results of the SWEEP runs are then transmitted to the chemist, who has the responsibility for analysis of the output, from which a determination can usually be made as to the nature of the erroneous or inconsistent information, and steps taken toward correction. Revised data are then supplied for subsequent SWEEP processing.

3. HADACOR. This is a routine for correcting the data and is executed after the analyst has inspected the results of SWEEP runs and determined what substitutions and modifications of data are required. When words are to be inserted or deleted, HADACOR performs the necessary pushdowns or other changes, as required; it makes replacements when only substitutions are required. It then reads in the SWEEP program again and initiates a processing of the corrected data.

4. SAND. SAND is a data formatting routine which at the same time compresses (to conserve storage space) the corrected information received from SWEEP and HADACOR computer runs. In addition, it performs various sorts, according to different sets of rules for the purpose of ordering information. SAND also assembles information and arranges it in accordance with the requirements of the structure search routine.

5. SQUASH. This routine performs the error-checking operations for the question data comparable to those which SWEEP performs for the disclosure data, or main disclosure file entry. There has been a deliberate attempt to make the question in many respects a match or duplicate of the file entry it seeks. However, the need to incorporate flexibility in the question has resulted in a somewhat different question format, with an increase in the amount of space required for question data over that specified for the same structure in the disclosures. It was possible to write SQUASH by making a modification of the SWEEP checking routine in order to accommodate the complete checking of the larger question format. (Attention of the reader is directed to earlier remarks on page 8.)

6. SQUAD. This routine is the data formatting and compression routine for the question which performs the same functions for the question which SAND performs for the disclosure.

The researchers lacked information with respect to desirable locations for incorporating screening operations. One option considered was the employment of screening against subsidiary data as a preliminary step to making the detailed search of the encoded structure files. An alternative proposal would intersperse the screening techniques at suitable locations in the computer program for execution of the chemical structure search. It was finally determined to follow the latter course, but in order to test the efficiency of the former approach, a task was assigned to a temporary employee (summer student) to write a test program employing the approach as described below.

The special screening routine performs the screening operation not on the disclosure data itself, but on other stored information which describes some general characteristics of the data contained in the disclosure file entry. In brief, the routine looks for the presence in the disclosure file entry of functional groups required by the question, although it does not ask whether they are present in sufficient quantity. It operates in much the same way as does a parts list which goes with an assembly program. There is a dictionary of terms which represents a listing of all of the functional groups contained in the encoded individual organic compound representations making up the file entries of the disclosure file. For each chemical structure represented in the file there is an unordered list of such terms, accumulated arbitrarily for each such structure; there is also auxiliary information such as bond connections between groups and detailed information in connection with rings or alkyl chains.

It was believed that many structures could be bypassed in the detailed search procedure if it could be determined in advance that the particular configuration did not possess in its listing one or more of the functional groups required by the question. For this purpose a table of 90 entries was devised where each one of 89 positions of the table represented uniquely one of the functional groups, and the 90th represented "all other" categories. 1/ The table, contained on magnetic tape, was addressed by the same unique file number which also referenced the detailed encoded file entry for that particular disclosure item. A table was made up for each of the entries in the file by inserting a "1" in the dedicated position in that table to indicate the presence of that specified functional group in the structure; and a "0" in that position indicated its absence in the chemical structure of reference. It should be noted that no provision was made in the table to account for multiple presence of any group in any one structure. For example if a particular structure contained three benzene rings, the table only gave indication of the presence of at least one benzene ring, without specifying the quantity.

A questioner who wanted to search for a particular structure or substructure had only to make a similar table showing in combination the presence of the groups contained in his "question structure". By an exceedingly rapid matching maneuver, similar to a simple overlay, he was able to pinpoint those entries in the file of encoded chemical structures which gave him a promise of success. He was therefore able to eliminate from his search a major portion of the file before he had called from the computer's store even the first file entry. Although this routine was run as an experiment against a table of entries describing the HAYSTAQ file, it was not used to obtain the results described in Section VIII. It could not have added significantly to the efficiency of the SEAC runs because the HAYSTAQ file of encoded disclosures was contained in a serial file on magnetic tape. However, successor programs are expected to embody some similar principles to those of the screening routine, where random access storage of the file entries will derive greater advantage from the use of such techniques.

VI. DESCRIPTION OF CHEMICAL STRUCTURE SEARCH PROGRAM

Techniques which have been developed for other problems were studied in order to determine their applicability to the development of an adequate chemical structure search program which would meet Patent Office requirements. Ray and Kirsch [8] wrote an exploratory computer program for an atom-by-atom chemical structure search; they employed a topological tracing approach, and some of their techniques influenced the HAYSTAQ development [9]. The patent examiner may, however, in a large number of instances have his needs satisfied by a search based on larger units than the individual

1/ SEAC words contain 45 binary digits each. There were a few more than 90 functional groups in the chemical structure information thus far encoded; thus, by storing only two SEAC words for each file entry and employing the rapid "extract" computer operation, a rapid scan of the file was possible.

atoms, e.g., functional groups generally recognized by chemists. Although the use of functional groups as the smallest units of structure in a topological system does not permit distinguishing the positions of substitution on rings or on alkyl chains, the patent examiner in general wants to find positional isomers since in the first instance they are taken to be equivalents.

For the majority of the cases of his interest, the patent examiner does not require the fineness of detail which the atom-by-atom search entails, and by accepting a larger unit of information than the individual element, the search program should be able to execute each individual search much more quickly. The basic vocabulary of functional groups is comparatively small, and was chosen to represent groups likely to be of use to the patent examiner.

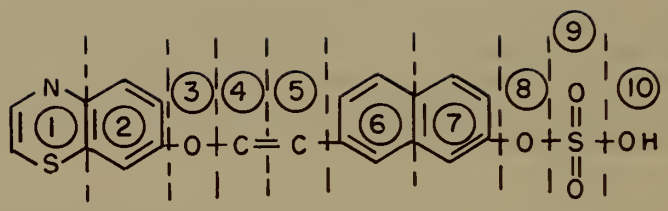
A close study was also made of the Norton-Opler system [10] for structure searching; this system deals with relatively large functional groups, in contradistinction to the individual atom treatment by Ray and Kirsch. HAYSTAQ's treatment lies between the two, and it was selected for the reasons discussed above. However, another motivating factor was its freedom from two recognized handicaps of the other systems: the relative rigidity of the Norton-Opler system and the relative slowness of the Ray and Kirsch system.

The data representing a complete description of the chemical structures are extracted from documents, as described in Section VII; they are then encoded and, after checking and rearrangement, recorded in a serial arrangement on a magnetic tape. The several magnetic tapes so compiled comprise the disclosure file of chemical structures to be searched. Each such entry in the file is an encoded chemical structure, and all such structures from one document are grouped together serially on the tape. Each such group of structures is followed by the group representing the encoded structures which were extracted from another document. Although it is convenient to group together all such encoded chemical compounds from a single document, the search program examines each chemical structure individually, and only in those cases where a match is found to a question does the program take note of the document number from which the chemical compound was extracted.

The arrangement from, for example, document number 37 would follow the following pattern:

Doc. No.	Disclosure No.	Data
37	1	(Here are listed all of the descriptive information with respect to the functional groups making up the first chemical structure extracted from document number 37.)
37	2	(Here is listed all data with respect to the second chemical structure extracted from document number 37.)
37	3	(etc.)

To compile the descriptive information for a structure, the structure is divided into its component functional groups, and each such group is assigned an arbitrary number which designates that group uniquely for any particular structure. The number is thus called a "designation number". (See Figure 1.)



1 -- thiazine
2 -- phenyl
3 -- oxy
4 -- methyl
5 -- methyl
6 -- phenyl
7 -- phenyl
8 -- oxy
9 -- sulfonyl
10 -- hydroxy

Fig. 1. The circled numbers are those arbitrarily assigned to represent functional groups.

A dictionary of substantive terms which represent the functional groups contained in the file has been compiled. In one-to-one correspondence with each name in the file is a five-digit alphanumeric code term which represents the substantive term in machine language. For a sample listing of a few such terms, together with their alphanumeric codes and their structures, see Figure 2. The five-digit alphanumeric code is the representation used for all mechanized processing of the information.

Name of Functional Group	Code	Structure of Functional Group
oxo	3COFA	= 0
oxy	3C10E	- 0 -
oxophosphino	3C001	P = 0
perchlorate	3C012	ClO_4
peroxy	3C006	- 0 - 0 -
perthio	3C050	- S - S -
phospho	3C004	0 = P = 0
phosphorous	3C018	- P -
phosphoryl		See oxophosphino
potassium	3C096	- K
selenium	3C01A	Se
sodium	3C0DA	- Na
sufonyl	3C104	S ⩵ 0 ⩵ 0
sulfoxy	3C00E	- S - ‖ 0
tellurium	3C01A	Te
thio	3C0C8	X - S - X
thiono	3C008	= S (when attached to a ring)
thiocarbonyl	3C032	≻C = S

Fig. 2. A partial listing of functional groups, together with their alphanumeric codes and their structures.

The encoded data for each file entry contains the following information:

1. A "package" of information for each functional group contained within that structure, where each such package contains the following:

   a. The encoded name of the functional group and its arbitrary number,

   b. A listing of the designation numbers of all other groups attached to the one specified and the type of bonding by which they are attached,

   c. (For alkyl and rings only) specific configurations for alkyl and detailed information about the nature of the rings;

2. Other types of information which are required by the search routine, some of which are used primarily for "housekeeping" purposes with respect either to screening operations or to accounting for matched or unmatched pieces in the search operation. See Section VII for more detailed explanations of data formats. The search routine provides for finding exact matches of structures -- structures which may be thought of as congruent -- or for finding a match for a fragment of a structure which may be embedded in a larger configuration. (See Figure 3.)

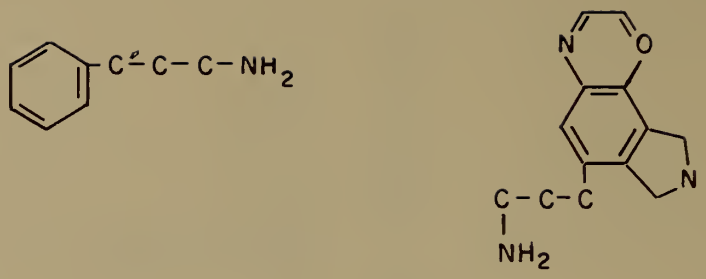


Fig. 3. The structure at the left represents the question and that at the right a matching disclosure file entry.

All questions to be posed to the file are explicit encoded representations of chemical structures and are in essentially the same form as the file entries. The question is asked of each file entry in turn, either (1) by attempting an exact match of the structure or (2) by considering the question as a substructure which may be found in larger configurations of structures contained in the file; the procedure is described below. The second of these alternatives is employed only when the questioner indicates that he would accept such answers. The numbers of both the document and the disclosure within the document are printed out for all cases where the question finds a match in the file.

The topological tracing procedure for executing each individual structure search is carried out by a sequential piece-by-piece match in the following way. (Appendix B is a general flow chart description of the structure search algorithm.) The functional group of the question which is listed first is arbitrarily selected, and an attempt is first made to find a match for that piece against any of the functional groups listed in the encoded structure from the file entry. (See Figure 4.)

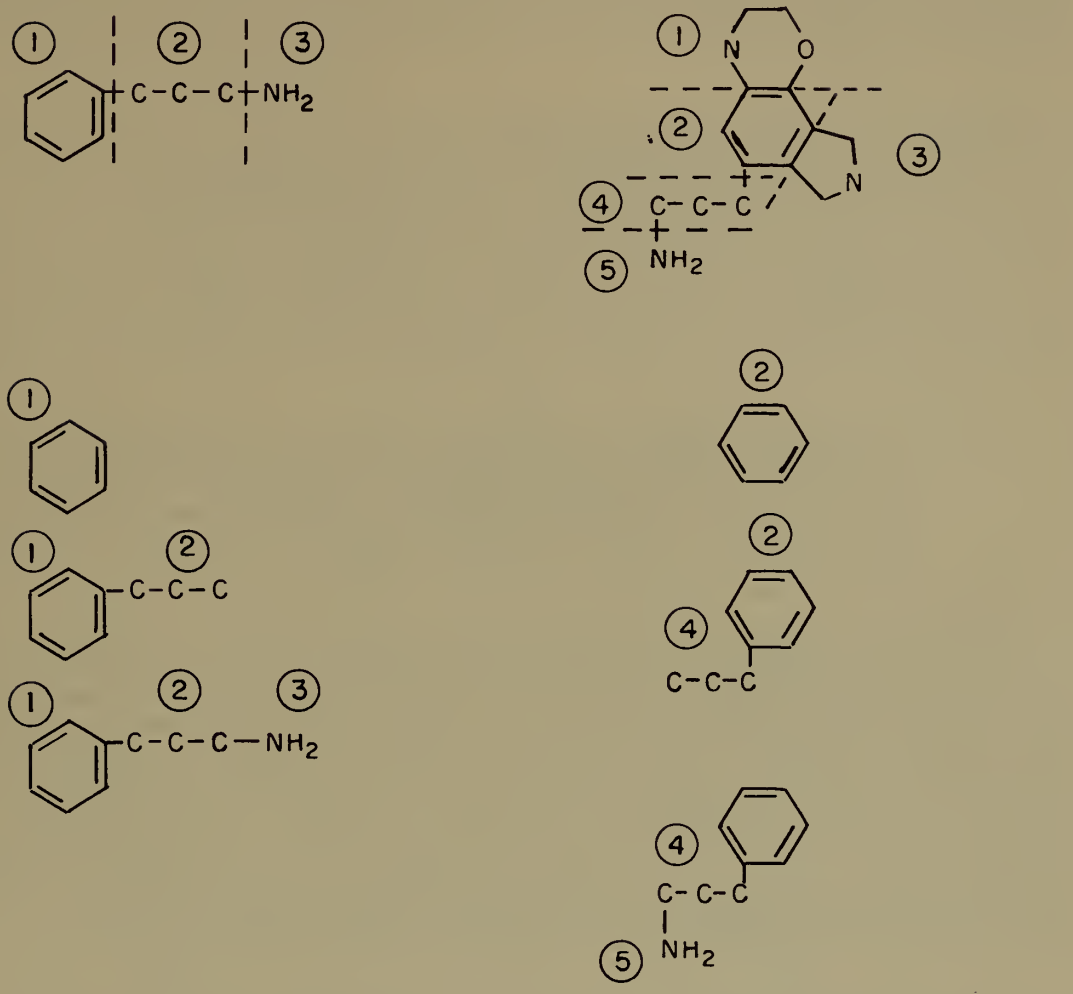


Fig. 4. The first line shows the structure for the Question and that of the Disclosure from the file entry. The next three lines show the result after successive matches of individual pieces. Encircled numbers are those arbitrarily assigned in encoding.

If a match is found, the two matching pieces are listed by name and number in a temporary table, as shown in Figure 5, and are temporarily crossed out from further consideration as matches for other pieces in the two structures. The search program then selects for its second trial the first functional group which is shown as a connection to the first-numbered question piece (in Figure 4 there is only one), and attempts to find a similar matching group attached to the first matched portion of the file structure. If it finds such a match, the bond connections of both question and file entry pieces are examined for identity.

Q Equivalence Table			D Equivalence Table		
1	Phenyl	2,	2	Phenyl	1, 3,4
2	Alkyl	1, 3	4	Alkyl	2, 5
3	Amino	2	5	Amino	4

Fig. 5. The tables of equivalence form the record accumulated during the matching operations for the structure shown in Figure 3 and Figure 4. The numbers following each substantive term are the designation numbers of the other pieces attached to it. Note that pieces of the disclosure structure which are not required as a match for the question are not entered in the Table.

The search strategy continues the matching operation from one functional group (or numbered piece) to another in this fashion until either a complete match or the absence of a complete match is discovered between the question and the structure as recorded in the file. If there exists a match, there have been obtained as a by-product of the matching procedure two tables of equivalent pieces, or functional groups, one for the question structure and one for the disclosure file structure. (See Figure 5.) At the option of the examiner, these may be printed out in order to assist him in the recognition of the topological equivalence; this may not be immediately apparent in very large structures where (1) the question structure may be drawn in a different way from the file structure, or (2) the question structure may be embodied in a very much larger file structure configuration, or (3) there are contained in the structure large Markush groups.

When any particular question piece cannot be matched, indicating an apparent mismatch of the two structures, the search routine attempts to trace another path through the network forming the file structure. As it "backs up" to try another route, appropriate cancelling of the tentative entries in the tables of equivalence takes place. Retrials will be attempted until it is discovered that there is indeed no match between the question structure and the particular file entry under examination. Identification of substructures contained within larger structures takes place in the same way as the matching of exact structures. With the provision for backing up and restarting when false trails through the network are followed, it is possible to find matches of like structures, regardless of whether the structures are drawn and coded identically. Because the designation numbers are arbitrarily assigned, the arrangement of the numbered codes which represent the functional groups are not likely to be always arranged in the same order, and frequent backups are expected and do occur, as attested to by inspection of intermediate results obtained by the computer in making structure searches.

A simple example has been chosen to illustrate the logic of the topological tracing algorithm. An attempt at this time to try to outline the complexities of generic search

problems overlapping with those of Markush groups and backup procedures, with the consequent records to keep track of all such interlocking operations, would only be confusing to the casual reader. It appears to be sufficient at this point to explain the general principle involved, and to recognize that there is a high degree of complexity in carrying out the details of the structure search in order to take account of all possible combinations of question and disclosure data.

Some Features of the Search Program

1. Patent examiners have frequent need to make several types of generic searches, and the computer program has been designed to execute such searches in accordance with the type of generic question which is asked. When a generic question is asked, the questioner may be satisfied by any specific embodiment of the genus. Thus, a searcher may ask that the attachment to a certain piece of the structure be a halogen atom, and he will be satisfied to find a bromine atom or a chlorine atom. On the other hand he may also be satisfied by a generic structure of the same scope as the question structure, such as a statement disclosing that the genus halogen is attached at that location. (See Figures 7a. and 7b.)

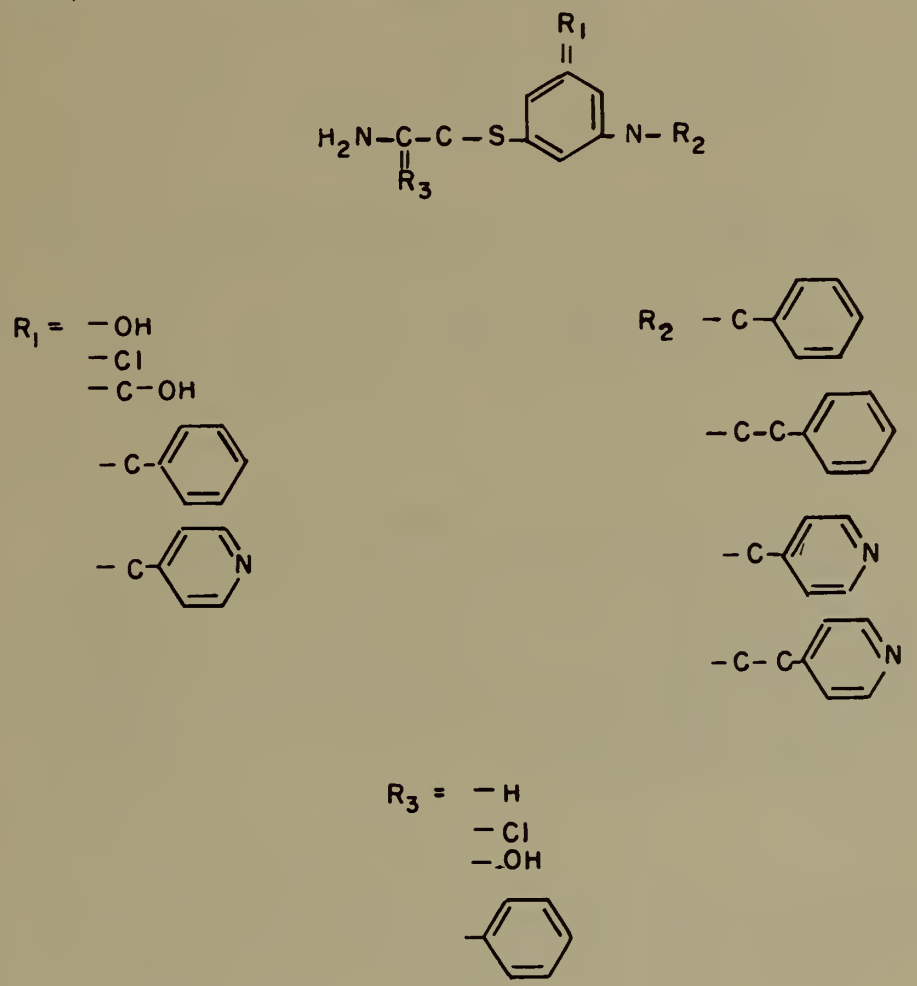
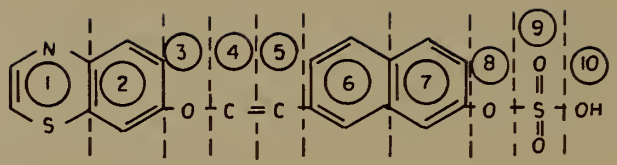


Fig. 6. An example of a simple Markush structure. Note that because of the Markush representation, 80 different compounds are represented by the one diagram.

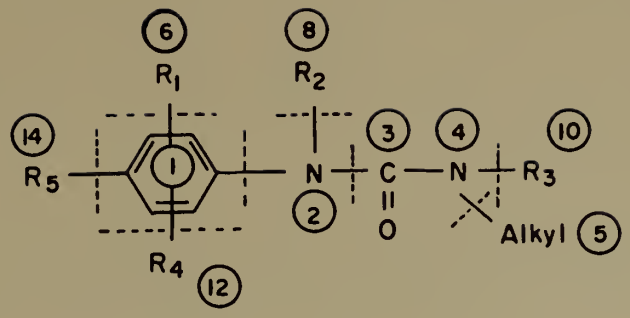


A Data	B Data
Benzthiazine	Ester - 7, 8, 9
Naphthaline	Alkene - 4, 5
Monosulfate	Acid - 9, 10
	Ether - 2, 3, 4

C Data - Topological Functional Group Description

1. Thiazine - 2(4)
2. Phenyl - 1(4), 3(1)
3. Oxy - 2(1), 4(1)
4. Methyl - 3(1), 5(2)
5. Methyl - 4(2), 6(1)
6. Phenyl - 5(1), 7(4)
7. Phenyl - 6(4), 8(1)
8. Oxy - 7(1), 9(1)
9. Sulfonal - 8(1), 10(1)
10. Hydroxy - 9(1)

Fig. 7a. An example of an encoded structure showing (1) the arbitrary numbering scheme, (2) generic concepts, (3) screening terms, as exemplified by the A Data and B Data listings, and (4) connections between the groups, together with the bond types forming the connections. Each of the four different bond types with which the program deals is denoted by a different number; the number identifying the bond type is enclosed in parentheses after the designation number denoting the connection to a particular functional group.



R_1 = – N.S. (6)
– Halogen (7)

R_2 = – N.S. (8)
– Alkyl (9)

R_3 = – N.S. (10)
– Alkyl (11)

R_4 = – N.S. (12)
– Chlorine (13)

R_5 = – N.S. (14)
– Halogen (15)
– Alkyl (16)
– O – Alkyl
(17) (18)

A Data - Large Substructures

Anilino
Urea

B Data - Generic Concepts

Halogen - 7
Halogen - 13
Halogen - 15
Amide - 2, 3
Amide - 3, 4
Ether - 1, 17, 18
Ring - 1

C Data - Topological Functional Group Description

1. Phenyl - 2(1), 6(1), 12(1), 14(1)
2. Amino - 1(1), 3(1), 8(1)
3. Carbonyl - 2(1), 4(1)
4. Amino - 3(1), 5(1), 10(1)
5. Alkyl - 4(1)
6. Markush - N.S. - 1(1)
7. Halogen - 1(1)
8. Markush - N.S. - 2(1)
9. Alkyl - 2(1)
10. Markush - N.S. - 4(1)
11. Alkyl - 4(1)
12. Markush - N.S. 1(1)
13. Chlorine - 1(1)
14. Markush - N.S. - 1(1)
15. Halogen - 1(1)
16. Alkyl - 1(1)
17. Oxy - 1(1), 18(1)
18. Alkyl - 17(1)

Fig. 7b. An example of an encoded structure showing the four features noted in Figure 7a, as well as the presence of Markush groups. The N.S. designation included in each of the five Markush groups in the diagram represents "no substituent".

A series of codes, known as the "A" data, is used to represent fairly large pieces of structure which are more inclusive than the single functional groups. Another series of codes represents generic - specific concepts and ring information, and this one is called the "B" data. The "B" data, as contained in Figure 7, are essentially screening information and appear in both the file entries and the questions to be matched against the file. Figure 7 shows two examples of how the structures are described by means of the two sets of screening data.

The A and B terms appear in both the file entries and the questions to be matched against the file. Those appearing in the questions must be matched by similar terms in the file entries, or the search proceeds no further in that particular structure. Thus, a preliminary examination of the A and B data indicates whether the file structure contains each of the structural concepts required by the question; the absence of any one of them in the disclosure file entry indicates that the detailed topological search would be fruitless.

The "B" data contain generic terms which in some cases refer to specific embodiments described in detail in the functional group description, or "C" data. (See Figure 7 and Section VII.) If the searcher, by means of the question which he puts to the file, requires that a specific type of acid be present, the program will first determine that the term "Acid" is present in the listing of B terms in the file entry. It will then determine whether the required type of acid is present by attempting to match the appropriate pieces in the detailed topological information listing. These pieces are those which are enumerated after the term "Acid" in the B listing, and the pieces so enumerated for the question B listing must find matches in those pieces so enumerated for the B listing from the structure description in the file entry. If there is no match of these "definitions" of acid) from the two structures being compared, then the particular acid in the file is not of the type required by the question, and the search terminates before it reaches the detailed topological tracing sequence of operations.

A searcher may only require that any specific type of acid group be present and the question will so reflect that requirement. In that case, when the B data are being examined from the file entry there is no attempt made to compare the specific definitions for acid. Instead, the file entry's definition is stored for later use in the topological tracing operations, and when that term is being considered in the detailed tracing of functional groups, the broad concept of acid is in effect substituted by the specific acid of the file structure.

2. A "Markush Structure", as the term is used in the U. S. Patent Office, is a generic expression; its use is a means of designating a synthetic genus which is defined by a listing of all members comprising the genus. The frequency of occurrence of Markush situations in patent applications is illustrated by the fact that in one week's issue of United States patents, more than two-thirds of the chemical patents issued contained claims in Markush form. Patent examiners have to deal with Markush structures which arise in claims, as well as in the reference material to be searched. It was therefore important to recognize the Markush problem and to make provision for searching Markush structures if the structure search program were to be of assistance to the patent examiner. However, major problems arose in developing in HAYSTAQ a general search program which at the same time could handle such generic structures expressed in Markush form. (See Figure 6 and Figure 7.) HAYSTAQ provides the ability to compare both Markush questions and specific questions with either Markush structures or specific structures in the disclosure file.

In the topological tracing operation, when the search procedure encounters a Markush group in either the question or disclosure data, the actual matching proceeds as though there were a variable connection for that group, where any member could constitute an acceptable match. If the Markush group is in the disclosure file, the question piece is in effect held constant and tried against each Markush member in turn, beginning with "No Substituent" (N.S.) which can constitute a valid match if the question so specifies. If a match is found in the Markush group for the question, the matching functional group from the list of Markush members is the one which is entered in the Disclosure Equivalence Table (see Figure 5) opposite the question piece as a match for it. If no match exists for the question among any members of the group, the search operation will encounter the flag which marks the end of the group and which signifies that no match was found for this question piece. The search strategy then calls for a backup, in order to try another match for the last previously matched question piece.

The Markush problem, an essential one for the U. S. Patent Office, [1/] was satisfactorily resolved for the structure search. Solving this complex problem was not without compensation; there were certain advantages associated with its solution. One advantage of the Markush data representation for the HAYSTAQ search is that it permits in effect a simultaneous search of many compounds.

The Markush feature has another advantage for the HAYSTAQ program with respect to data preparation. Compounds in a document which have many elements in common may be grouped together and the common portion represented as the invariant part of the diagram, while the diverse portions are represented as the variable part of the diagram. This enables a compression in the amount of information to be stored for the group of structures so represented, and at the same time enables many structures to be searched by a single matching operation of the question against the Markush structure. It should be pointed out, however, that the diagram, while appearing to contain a variable constituent, in reality is a representation of a number of different compounds: specifically, the number of compounds represented is equal to the product of the numbers of members in each of the Markush groups. It is possible to include "No Substituent" as an acceptable member of a Markush group. This signifies that the structure is made up of any one of the members of the Markush group listed as shown at that position, or that a hydrogen atom attached to a nuclear carbon at that position has not been replaced. (Normally, HAYSTAQ ignores hydrogens attached to nuclear carbons.) The "No Substituent"condition is usually listed as "N.S." (see Figure 7).

There are now 2,400 encoded structures in an experimental file contained on magnetic tape. A large proportion of these are Markush structures, each one of which may represent many different specific compounds. The tape actually contains representations of more than 162,000 individual compounds, although to search the entire tape it is only necessary to try a match against the 2,400 individual entries. It is difficult to overemphasize the importance of the ability to search many compounds by means of a single file entry, or on the other hand to pose many questions (in the instance of a Markush question) by means of a single encoded structure. In addition, the saving in data storage is not inconsiderable.

1/ Other organizations admit they have the same problem, even though it is referred to in different terms.

3. Both rings and alkyl groups received special treatment because of their generic character and the frequency of their occurrence in documents dealing with organic compounds.

a. Provision was made in the input data formats for including additional descriptive information in fixed fields with respect to alkyl. The use of "alkyl" as a generic expression is made more specific in each case by including the number of carbons in the chain and by identification of specific configurations. A special subroutine was written into the program to execute the details of such searches.

b. Detailed information was also included in the data formats to describe rings, e.g., homocyclic or heterocyclic, with further breakdowns on these two categories (see page 38). Again, a special subroutine permits searching for rings in the required detail.

4. Screening operations were carried out as a preliminary step prior to initiating a detailed topological tracing, and they were also inserted at strategic locations in the program after the detailed search had begun. Even with the assistance of computers, a considerable amount of time is required to search through a very large file by means of a serial approach. In order to shorten the search time, there are included with the encoded structure two general types of screening information which are scanned before the search progresses to the detailed topological tracing. Experience with use of the program indicates that in the majority of the cases an examination of the screening data revealed a mismatch and eliminated the necessity of entering the more time-consuming piece-by-piece matching, which has been described above and is illustrated in Figure 4.

The importance of the screening by means of the A and B terms was discussed on page 20. An additional screening effect is obtained by examining the B data of the question in order to determine whether the functional groups referred by its B generic terms will find matches in the disclosure among the functional groups which are the referants for the disclosure B terms. The functional groups in this sense may be considered specific "definitions" of the generic B terms. (See Figures 7a. and 7b. and Section VI.1.

It may be observed that if a searcher is interested only in making a very general type of search, he may search only on the A and B terms.

This coordinate indexing type of search could be performed very rapidly, even for extremely large files, because only a few computer instructions would be required to execute the search for any one entry; thus the search time would be almost limited to the input time for the data plus the printing time for the document numbers containing answer A few such searches were made by HAYSTAQ, but in general the examiner is interested in obtaining the results from more detailed searches.

Other types of screens were employed throughout the search routine. For example, the number of functional groups to be found in the structure is contained in the heading information which precedes the remainder of the encoded data defining the structure.

If the minimum number of such groups required for a question structure to be matched is greater than the number shown to be contained in the file structure, there can obviously be no match and the search routine progresses to the next file entry. As the search routine selects each functional group from the disclosure file entry in the network tracing, the number of its connections to other groups is investigated in order to determine that such a group has at least as many other groups attached to it as the question group requires. These, and other screens, thus have the potential for terminating the search at various levels of data examination.

VII. DATA PREPARATION

HAYSTAQ presumes that the file data are in a suitable form for immediate searching; this form is a highly compressed, ordered, preprocessed file of information, arranged according to a definite pattern which is conducive to early recognition of a match or a mismatch. The purpose of this Section is to describe how the data reach that highly stylized format from their original status where they first represent some significant part of the document content.

Documents containing chemical information are analyzed by chemists, who extract the organic chemical structure information from the documents, draw the structural diagrams, and write, in coded form, the descriptive information making up the first formatted chemical structure data. The data in this form are then ready for tape punching operations.

The data extraction calls for a high level of skill from the chemists who analyze the document content. Not only must all generic relationships be recognized and defined, and Markush groups properly recorded, but implicit references in the document to organic compounds must be recognized and made a part of the file entry. This procedure would preclude analysis of documents by a non-chemist; it would also preclude such data extraction procedures as the underscoring of pertinent information.

The data for the program are arranged in three groupings, except for appropriate heading words; the latter contain housekeeping and other relevant information for programming requirements. The three groups of data are the A and B data, which have been previously described, and the C data, which contain the detailed descriptions of the functional groups. As a carryover from stage one of HAYSTAQ and in order to denote the relationship of these data to the category 3 information in that system, these data are commonly referred to as the 3A, 3B, and 3C data. The 3C data describe in detail the nature of each functional group and specify connections between functional groups. The 3B data describe generic-specific relationships and ring information, and the 3A data reflect larger groupings of the 3C data. The 3A and 3B data are employed for screening purposes, and the 3C data are employed in the detailed tracing operations.

Experience with manually prepared test data for program debugging operations for HAYSTAQ indicated a high incidence of error, despite the exercise of extreme care in the initial preparation of the data and its subsequent careful checking by professional personnel. Experience with the use of small amounts of hand-prepared test data pointed to the necessity for carefully thought-out data preparation routines for computer execution. Such mechanized assistance in data preparation was particularly desirable in two areas:

1. Detection and correction of such errors as lend themselves to discovery by machine methods, and
2. Compression, ordering, arrangement, counting, and other manipulation of unprocessed data in order to assist in obtaining the final formatted file to be searched by HAYSTAQ.

Two data checking routines, SWEEP and SQUASH, for disclosure and question data, respectively, were written by Robert T. Moore to accomplish the first of the two objectives; they are described as Part I of Appendix C. Part II of Appendix C describes the data preparation, error correction and record-keeping procedures, as well as the routine for ordering, sorting, arranging, and compressing the corrected data: these routines were called SAND, SQUAD, and HADACOR (see page 9). The latter routines were written by Catherine E. Lester, who set up the elaborate system of records which the nature of the operation to be undertaken required. She also supervised the tape punching operations and the training of the paper tape punch operators. In addition,

she instituted measures for the exercise of quality control on the preparation of the paper tapes and on the flow of corrections issuing from the analysts in response to errors detected by SWEEP and SQUASH. Her observations on the nature of the errors encountered, the patterns into which they fell, and remedial measures for the avoidance of errors have been particularly helpful to the entire project.

The coded but unprocessed data were punched manually on paper tape by means of a Flexowriter, and this punched paper tape formed the initial input medium for the data when they were read into the computer by SWEEP. The efficiency of any program depends upon the accuracy of the data it manipulates, regardless of how sophisticated the program logic may be. It was not possible to program the computer to check for every type of error, especially those resulting from tape-punching operations. Every precaution was therefore taken to ensure the preparation of error-free paper tapes.

It is possible, but not probable, that two people punching tapes will make the same type of mistake in the same place; thus, all data were typed by two different people on different flexowriters, and the two resulting tapes were compared by a mechanical device which indicated any discrepancy between the tapes. The correct tape was verified, and a notation of the errors was marked on the incorrect tape in each case. At the end of the comparison, the tape with the fewest errors was duplicated with corrections in order to obtain an error-free tape for the computer operation. The corrected tapes were joined together in sequence by disclosure number and read into SEAC by means of a Potter tape reader. They were then processed by SWEEP.

All operators were given both verbal and written instructions; these were later revised to include more detail after experience indicated the need for amplification. The revised instructions also included information as to the types of descriptors encountered in the data, the minimum number of words required for each type of descriptor, the signal for the end of a descriptor, and the proper arrangement of descriptors and flag words in each section. The operators were encouraged, both verbally and in the written instructions, to question any manually coded data which appeared to be erroneous rather than to risk punching incorrect data.

Successful culmination of the work on this project required the utilization of several discrete classes of skills. The analyzing and encoding of the technical information content of the documents was exacting work which required the skills of professional chemists. Their skills were again employed in the detection and correction of certain classes of errors which occurred in the original data forming the input to the question and disclosure files. (See pp. 9, 23, and 36 and 37 (Appendix C)

VIII. RESULTS OF COMPUTER OPERATIONS

All of the computer operations were carried out on SEAC. The total file accumulated consisted at final count of 2,393 entries taken from 185 documents. Because of the Markush feature, which permitted the representation of many structures in a single file entry, the total number of individual chemical structures specified in the file was 162,000. Some disclosures were found to be too large to be accommodated in the amount of memory space available for data storage, and had to be eliminated from the initial file, with the result that not all disclosures extracted from the 185 documents are included in the file.

An initial question file of 385 organic structures was compiled. The questions were devised to insure that as many different types of searches as possible were run. Some questions were designed to test the Markush features of the search program, other to test simple specific structures; some were designed to test cyclic structure searching, others to test the accuracy of the program in searching very large and complex structures.

Other questions were directed to searching structures with relatively large numbers or rings and/or alkyl groups, and still others to searching structures which had long lists of A and B terms, with the facility for matching definitions in the case of some of the B terms. Six broad generic questions received, as expected, a large number of responses (e.g., find all structures in the file containing methanol); the greatest number of answers resulting from such a search was 128.

Some of the questions were found by SQUAD to require more machine storage space than was available, and 17 such questions were eliminated for space considerations. Five additional cyclic questions were not run because an attempt to run the first one indicated a mistake which still existed in the routine for making cyclic searches. Although the correction measures were not complex, there did not remain enough storage space to accommodate them. The necessary correction steps were therefore documented for future guidance.

There were altogether a total of 363 questions which were put to the file. The 363 questions posed against the 2,393 disclosure entries in the file represents a total of 868,659 individual matches of questions against disclosures, disregarding of the Markush feature. Taking into account the Markush structures, there was a capability of matching 58,806,000 structures; however when one member of a Markush group provided a structure which satisfied a search question, the remaining possibilities were not tried. There were no statistics kept on the number of such trials of different possibilities in Markush groups which did not result in answers to the question.

Before the computer runs were made, cards had been compiled which contained the structures represented in the various file entries on magnetic tape. From the information contained on the cards, it appeared that the search program, if successful, would be able to elicit at least 573 answers from the file in response to the 363 questions posed against it. In reality, 538 additional answers were detected by the search routine, all of which were proper answers. These had not been identified as answers from a manual search of the card file entries. In a sense it may therefore be said that 1,111 answers could be expected from the computer runs. Of these expected 1,111 answers, 839 were retrieved by the computer, and 272 were not. In the following analysis of the failures to retrieve, it will be seen that such failures are attributable to a variety of causes.

The answers which might have been expected but were not found represented almost 25 per cent of the total number. Analysis of the results showed that there were three general categories of "lost" answers:

1. Apparent losses (56) which were not errors:
    a. 24 represented answers to questions which had not been put to the file (for considerations of storage space and the cyclic situation),
    b. 15 errors had been made in the manual search of the file, and
    c. 17 disclosures containing the expected answer had not been entered in the final magnetic tape file.
2. Human encoding errors (216):
    a. 154 were errors made in connection with encoding certain types of information such as screening information in A and B data, ring details in 3B, etc., which could have been eliminated if machine encoding of these types of information had been employed.

b. 29 were substantive coding errors, for the most part errors in dictionary look-up of terms, and

c. 13 were miscellaneous errors of a character not provided for by the error detection logic in the SWEEP and SQUASH routines. 1/

3. Deficiencies of various kinds in the search routine (4); these included logical errors or omissions and housekeeping operations.

An additional 16 of the lost answers have not yet been explained. It may be that some of them are attributable to machine error, since reruns of some of the previous runs produced answers which were not found in the first instance. Others may be due to obscure errors in the data, or to a further deficiency in the search program, either due to a logical error or to errors in the housekeeping operations involved in keeping track of the tracing operation in some of the very complex situations, e.g., combinations of backtracking with generic searches of multi-membered Markush components of a functional group.

The average length of time required to search the entire file in response to one question was approximately six to eight minutes. Many searches required no more than one-and-a-half to two minutes. This is a relatively short search time, particularly in view of the following considerations: (1) The file contained approximately 2,400 entries, representing 162,000 organic compounds (because of the Markush feature). (2) SEAC, which was the first operational stored-program computer in the United States, is naturally much slower than modern computers, both in internal processing and in input. (3) For the purpose of ease of testing the basic logic, the search was conducted on a serial basis, with the effect that the entire file of disclosure entries was read into the computer for the search of any one question. This situation would certainly be improved in a production-type operation by parallel processing and by the development of a variety of file organization schemes.

For the majority of structures contained in the disclosure file, the topological tracing portion of the routine was not entered at all for any particular question. Instead, the search was terminated after the screening operations were conducted against the A and B terms; such screening revealed the futility of further examination of the structure contained in that file entry. The search time was much longer for those cases where the structure in the file was very similar to that of the question, and an apparent match was indicated during the execution of the first portion of the topological tracing routine.

Such patterns of behavior were of interest to the originators of the system. Of particular interest, of course, were the reasons for failures to find anticipated answers to questions. As discussed on page 25 , more than half of the failures were occasioned by the subjectivity of the analysis procedures employed which permitted the individual interpretation by the analysts with respect to the A and B terms to be supplied. During the screening operations preliminary to the detailed search, failure to find any such A or B term which is required by the question results in a termination of the search of that particular file entry. Many such searches were thus terminated because of the absence of A and B terms called for by the question, even though an answer in fact existed and would have been found had the topological tracing portion of the routine been executed. In other words, the effectiveness of screening is highly dependent upon the accuracy and consistency of analysis with respect to the assignments made to the A and B terms.

At an early stage in document analysis and file preparation, arbitrary dictionaries

1/ Acceptable procedures to cope with such errors need to be developed. (See page 37, Appendix C.)

of A, B and C terms (intended to be as inclusive as possible) were compiled. The chemist who analyzed the document and extracted and encoded the structures therefrom also selected from the dictionary all of the A and B terms which he thought applicable to any particular file entry. In many instances, the subjective nature of the decisions by which A and B terms were selected was such that question and file structures which were in fact identical, but which were encoded by two different persons, contained different screening terms. Thus,when a file entry lacked a screening term which the question demanded, that file structure was rejected as an answer even though the structures of question and disclosure were identical.

On the basis of the number of trials run,the search algorithm appeared to be adequately effective and efficient; testing brought to light minor deficiencies with respect to bookkeeping operations. In these cases, correction methods were outlined for future recognition; space considerations did not permit their insertion into the program at this time. The experience gained in data preparation and data manipulation was very valuable for the sakeof future progress in these areas, as discussed in the next Section. The effectiveness of the screens employed seemed to indicate that it would be profitable to explore the development of additional ones for future operations. And, finally, the fact that almost as many unanticipated answers were found as those anticipated by prior manual search was extremely encouraging from the standpoint of search adequacy and the development of a feeling of reliance on the search results. The intuitive feelings of the investigators that humans would inevitably miss many references which a comprehensive mechanized system would retrieve were verified by the results. Past experiences of the Patent Office with the mechanized steroid search had previously reinforced these feelings.

IX. LESSONS LEARNED AND THEIR EFFECT ON PLANNING

1. Experience with the large manual effort in data preparation illuminated the necessity of providing for mechanized data preparation in all possible areas of future operations. Not only was the manual encoding believed to be much slower and more costly than mechanized execution of the same tasks would have been, but the incidence of error was distressingly great. Machine error in those cases would have been almost nonexistent. It is not known how much time was spent and how great the cost in eliminating those errors; this is true from the standpoint of human effort expended manually as well as for computer runs of HAYSTAQ, since no separate record of additional effort spent on these operations was documented. Again, the manual effort of preparing the A- and B-level terms was by no means insignificant, and it has already been pointed out that numerous failures to find answers to questions were occasioned by the human differences of opinion in preparing a comprehensive set of such terms, both for question and disclosure structures. Automatic preparation of such comprehensive sets would be feasible once chemists provided the specifications to be embodied in a computer program for accumulating such terms. Such a machine-created set of terms would have the advantage of consistency, whether or not the sets were always comprehensive, and thus failure to find answers for the inconsistent cases would not have occurred.

2. There were some redundant entries in the file in the following sense. When a document was analyzed, all of the organic chemical structures from that document were extracted and encoded. When one of the same structures was later encountered in another document, it was again extracted and encoded. There was no effective means for recognition of a prior entry into the file. No attempt was made at that time to number these compounds uniquely so that a different person, or even the same person who had previously encoded the same compound, could identify a prior entry and make use of it, thus avoiding duplication in diagramming and encoding. Recognition of this handicap has influenced plans for future work, as discussed in Section X. A different approach to file organization and the storage of the chemical structure information in the file is presented in that Section.

3. Although there is promise of future benefits from some of the ambitious syntactic analysis and other linguistic research now going on, at this time there is no better way available for document analysis than human recognition of the items to be entered into the file. This statement is particularly true in those cases where the original text makes only implicit references to structures, and the structures are nowhere explicitly stated in the document. However, once the implicit reference to a structure in the document has been recognized and its diagram drawn, it is desirable to have a quick and economical method of entering the structure into the file. This consideration is one of the reasons why a linear notation [11, 12] is being employed for future file preparation.

The linear notation method of input requires only one manual operation after the initial enciphering of the structure from the diagram: that of typing the notation on either punched paper tape or punched cards. This subject is discussed in greater detail in the next Section. On the basis of ciphering trials using the Hayward notation [22], it appears that this method of input should result in greatly reduced time and cost of file preparation; there will be fewer stages of manual handling of data, with a decreased cost in initial data preparation. Not the least reduction will accrue from elimination of the elaborate error detection and correction procedures at each stage of manual handling.

4. The power of the screens employed was vividly demonstrated in this first structure search program by the rejection of major portions of the file before the topological tracing routine began. For that reason, it is desired to provide many more screens, and screens of a diverse character, for future operations [25]. Such screens should be selected so as to be as mutually independent as feasible, i.e., without overlapping of coverage. An implicit method of screening can be achieved by an arrangement of sub-files, each ordered with respect to content (so that subfile need not be examined in any way if none of its contents can be applicable). Such files will be discussed in Section X (see page). On the basis of the limited experience obtained from computer runs to date, it is believed that the provision of such a battery of screens may demarcate the difference between efficient or inefficient searches, with the related questions of economic feasibility. The provision of a sufficient number and variety of screens becomes even more important from the standpoint of two considerations that relate strongly to the need for efficient searches:

(1) the anticipated large increase in the number of file entries, and

(2) the need for progression to finer detail in the searching, i.e., a complete atom-by-atom match of the structure, with a consequent increase in search time for each search.

5. It is expected that the only system which could meet the total U. S. Patent Office requirements would be an extremely large-scale operation characterized by a voluminous file. The serial nature of the file has thus far been tolerated as a research expedient in spite of the time required to search all entries, one at a time. This will become unacceptably burdensome as the file increases in size [25]. Therefore, a significantly different file organization for future chemical searching is contemplated. That is a subject of discussion of the next Section (X).

6. Although the ability to make the structure searches would be valuable to the patent examiner, it does not fill his total requirements. Therefore, it is anticipated that future information files will have associated with the structure additional information such as chemical and physical properties, usages, chemical reactions, as mentioned in the next Section. In this connection, it is also recognized that there is need for an extension of search techniques to inorganic structures. and future research is intended to include work in that area.

X. FUTURE RESEARCH

Information from chemical patents formed the corpus of the mechanized file for HAYSTAQ and organic chemical structures were selected as the subject for mechanized searching, to the exclusion of other document content. Chemical structures constituted an initial research area of manageable size for the first fairly large-scale computer operations, although it is intended to include in later comprehensive search programs the capacity to retrieve other kinds of chemical information. Of particular interest, as noted above, is the extension of the retrieval programs to inorganic structures. It is also expected to include physical and chemical properties of compounds, process information [4], uses and other auxiliary information such as biological effects and bibliographic references. The feasibility of indicating, within a file of chemical structures, the presence of other types of information is being investigated.

One of the principal areas of concern at the present time is the improvement of data preparation for HAYSTAQ. Improvement is desirable from several standpoints: cost, time, kinds of human labor demanded, and, above all, reliability and consistency of file content as reflected in the elimination of human-generated errors. It would be advantageous to represent a chemical structure by a procedure that is less dependent on subjectively derived groupings than the procedure that was employed in HAYSTAQ. It would also be desirable to reduce the number of times the data must be handled.

Such a procedure would reduce the sources of human errors. The employment of a linear notation [11,12] to describe the structure appears at the present time to offer an attractive approach for satisfying these desiderata. If at a later time some other method, such as optical scanning, for example, should prove to be a more efficient and reliable means of entering chemical structure information into the file, the current use of the notation system would not preclude a later shift to a different mode of input.

The linear notation system originated by H. Winston Hayward, of the U. S. Patent Office, is being investigated for the input of structures to the file [11,12,22]. This notation system was designed with machine processing requirements in mind. It offers a means for a unique and unambiguous representation of every organic structure. The Hayward system is relatively easy to learn, and individual ciphers can be written quickly. The uniqueness of the cipher, its ease of learning, and the speed of enciphering are all advantages over the encoding system previously used.

A group of college students was employed during the summer of 1963 to encipher organic structures in order to have a file of linear structure notations which could be used for testing the validity of such representations. More than 60,000 ciphers were written by the group representing structures taken from Index Chemicus [26] and the Revised Ring Index. [27]. The students, with varying backgrounds in chemistry, learned to encipher using the Hayward system with an acceptable level of competence after an initial learning period of about two weeks; the average length of time required to encipher a structure was about two minutes, where each structure represented a single specific compound.

Although the linear notation is a convenient form for input to the computer, without a transformation it can be used only for "dead match" structure searching. Several algorithms have been written to transform the linear notations to a tabular array in which individual atoms and bond connections are shown [13]. Algorithms have also been written for the automatic generation of screening information and others are under development for error detection and correction.

Assuming that experience with the Hayward notation system verifies its premise as an input technique, then the ciper will be stored as the structure description in a file entry, along with other information pertaining to a particular compound. Associated with each

cipher will be a unique identification number for reference to the structure and to the associated information. It is expected that the unique identification number will provide a convenient access both to the structure itself and to the related information, which may be stored in subfiles in different physical locations or even on offline microfilm equipment or in file folders. The number will thus provide a means of random access to the structure and all of its associated information.

A complete listing of information about each structure will be contained in the master file entry, but it is contemplated that subfiles will be assembled by the computer with provision for accessing them without initial reference to the master file. It is expected that the programs for file updating and some of those for searching will be of the so-called list-processing type, and that some of the files themselves will be strung together by means of list-processing techniques [14,15]. The subfiles in many instances may be homogeneous in content, and be based for the most part on discrete classes of information. In addition to the creation of files based on various types of non-structure information, it is contemplated that special chemical structure files will also be accumulated, based, e.g., on molecular formulae or on the occurrence of specific substructures. Such subfiles would provide implicit screens to file content. There is need for more operational experience with such arrangements of files where large volumes of data are to be searched. A serial file arrangement does not permit sufficient distinction to be made between large masses of non-pertinent information and the relatively few items in the file which are of interest to the searcher. When a direct access capability is permitted by a different file arrangement which provides for random access, subfiles may be accessed according to the kind of information contained in them, regardless of where they are located. One strong advantage accruing to the user of information so arranged by subfiles is the relative ease of obtaining periodic hard-copy printouts of compendia from the specialized subfiles. Such printouts have value for manual searching and constitute a bonus from a mechanized system which is available for little additional cost. Comprehensive compilations of updated and current information might be particularly useful where a given type of information is sought frequently, as in areas of new technology where there is increased activity, e.g., the areas of hormones or of antibiotics.

Given the existence of a large file of complete information, supplemented by many specialized files or subfiles, and the necessity of obtaining associated information from several of the special files, it becomes necessary to provide a "key" for cross-referencing in order to bring together, when required, information existing in separate files. Such a key might be provided in the form of a special index file, which would not only relate the information in the mechanized portions of the file, but would also tie together pertinent information stored on offline equipment, e.g., microfilm libraries and hard-copy records in numbered folders. It is expected that the master file would not normally be used for the searching operations, but that it would remain the repository of the complete set of data for each structure in the mechanized file, and it could be accessed from other locations by means of the unique identification number of the structure. Cross-referencing techniques among various kinds of associated information in a file organization somewhat similar in nature have been discussed by Prywes and Gray [14], and others.

Certain types of information do not easily lend themselves to unambiguous classification. This is particularly true for information about the properties, behavioral characteristics, and other attributes of chemical compounds, as contrasted with information that describes their structure or atomic configuration. At this time, therefore, it is contemplated that labels, or "tags", will be supplied to designate certain categories of information found in a document to indicate the presence of some kinds of non-structure information. For example, a tag might reveal the presence in the original document of biological information about the biological properties or commercial utility of a compound.

These labels may be viewed as categories which, on a pragmatic basis, may be divided into subcategories for the purpose of making more efficient searches; however, no claim is being made at this time that adequate methods exist for handling the classification of this type of information. In some cases, this problem arises from a lack of scientific discipline as to the nature or significance of the information. In other cases, classification is difficult because the basic ideas in a document are completely woven together and not sufficiently precise or concrete to permit useful classification.

A guiding philosophy of the present work and the planned future work is the design of a modular approach, for both files and computer programs. Data or information will in many cases form modules. Computer programs will be developed as building blocks or modules, and will be accumulated in as many different kinds of packages as there is need for different search patterns. It is expected that experience obtained through use of the system in searching will indicate trends toward such patterns and that there will come into being a library of "canned" searches. For example, a chemical structure search might be combined with a "use" search where compounds containing such structures have been used for a particular function. A modular system organization would permit the replacement of some routines in the system with others which execute different functions, as needs change, or the substitution of modules with modified or improved versions; and it would also permit a constant shift and change of program emphasis and data, both of which are required at the Patent Office in order to respond to changes in the technologies with which they deal. The modular approach to computer program development will permit the assembly of modules or building blocks, as needed, to fulfill the requirements of making individual searches on demand.

It will be necessary to develop an Executive Routine to accumulate program packages which have frequent use for storage in the program library and to compile search programs on an ad hoc basis when unusual kinds of searches are required. The Executive Routine must also act as the control for all operations; it must interpret each question and direct the execution of the search requirements of the question; and it must also direct communications among the various input-output media. In addition, it must provide for pinpointing deficiencies in the existing search techniques in order to point the way toward desirable amplifications of the system. A large system for carrying out diversified operations will require other automatic programming aids, as well.

For example, the Executive Routine will have to communicate in at least three languages: the direct machine language itself, the specific programming language (compiler or assembly system) employed on the particular computer being used, and a special information processing language (IPL) for chemistry which will permit chemists to express their requirements in terms taken from natural language, even though in a somewhat stylized manner. Such a set of terms should remain open-ended to allow for expansion of the information processing language. The chemical IPL development will be intended for use with HAYSTAQ, although its use need not be limited to HAYSTAQ. It can be universal and might be used to communicate between different organizations who are interested in processing the same type of information. The chemical IPL will thus act as a translator between the outside world of the chemist and other scientific personnel, on the one hand, and the inside world of programming languages and machine operations, on the other.

In certain areas of the present research, new approaches to problems must be sought and explored. Some of the required research tasks which have been discussed in this Section are largely theoretical at the present time, and little or no concrete work along the lines discussed has yet been undertaken. However, in making plans for an improved HAYSTAQ, they offer promising paths for investigation. Although much of the work discussed here is still in the planning stage, the linear notation systems for representing fully defined organic structures and certain types of Markush structures are essentially

complete [11 12]. Work is currently going forward in developing the notation system to cover inorganic structures [23,24], other types of Markush structures, polymers, and structures with partly undefined connections. Several algorithms for cipher manipulation have been programmed and run successfully on the NBS PILOT Information Processor. Some work on file organization and on screening techniques has been initiated, although both such projects are still in preliminary stages of investigation.

Two different algorithms are being investigated for structure search routines and will form program modules of the completed systems; one of them is the topological tracing at the atom and bond level, as described for functional groups, and the other is a matrix manipulation method developed by Dr. Edward H. Sussenguth, Jr. [16], In the latter case, a more powerful technique is required for the Markush cases which cannot be handled by this method.

While the actual programming is being written in the Pilot language for execution on that computer, the evolving system is believed to be computer-independent, at least through the flow-chart stage, with respect to the system design and the description of the logical steps required to carry out the details of the system.

XI. OTHER RESEARCH RELATED TO HAYSTAQ

The last few years have seen increased activity in the exploration of mechanized means of searching chemical information. Chemical Abstracts Service (CAS) in particular has been engaged in developing mechanized search routines for specific well-defined determinate structures [17]; for several years CAS has been accumulating a file of chemical compounds by means of the Dyson (IUPAC) linear notation. [28] In this research they have been supported by both the National Science Foundation (NSF) and the Army Research Office (ARO), in addition to the support for research provided by the American Chemical Society.

The Army Research Office has embarked on an extensive project to develop a mechanized system for searching chemical information.[1/] It has the responsibility for carrying out this activity for the needs of all the Department of Defense. To assist them in achieving their purposes, they have allocated responsibility for separate functions of the activity to different Army laboratories, and have sponsored several research projects with other organizations, including the National Bureau of Standards.

The Food and Drug Administration (FDA) has stated that it must be concerned with chemical structure searching in order to discharge its responsibilities properly. Because of its other requirements for mechanization and its already overburdened staff, it is considering the prospect of either buying a system if a satisfactory one can be found or of contracting for its design. [18]

Various chemical, drug and pharmaceutical, and petroleum companies are developing means of searching chemical information for their own purposes, but a great deal of this type of activity is retained within the company as proprietary information. From the limited amount of information which is available from such activities, it appears that none of these systems has the power which the U.S. Patent Office would require for its searching; this is certainly understandable because most companies have areas of concentration with respect to their marketed products, and none has a need for the complete coverage

1/ This effort is better known as the Chemical Information and Data System or CIDS project under the STINFO (Scientific and Technical Information) program.

for which the Patent Office has responsibility.

The Badische Anilin- & Soda-Fabrik, AG, a German chemical firm, has for several years been interested in the possibility of mechanized searching of chemical structure information [19]. One of their employees, Dr. Ernst Meyer, has invented an optical scanning device for the purpose of putting into the file by this means a portion of the chemical structure diagram: the part which can be obtained by connecting points in a grid to form lines. Bond connections and element designations must be punched on additional cards or tape and merged with the optically scanned information by an appropriate computer program to form the complete structure.

The U.S.S.R. has announced that she will engage in a vast program of chemical research, assisted by mechanization. Little is known of Russian activities in recent years, although fairly extensive reports were made a few years ago of several chemical information storage and retrieval projects [20].

Finally, the Modern Methods Committee of the National Research Council (NRC) has been investigating various methods of representing chemical structure information as input to a mechanized file [21].

The researchers who are responsible for the development of HAYSTAQ are keenly interested in the worldwide ferment of activity in mechanized chemical information searching. They must and will continue to try to keep abreast of progress by others in this area in order to build on what others have done, as well as to keep in mind considerations of compatibility for the purpose of possible exchange of information and data. At the same time, HAYSTAQ is directly concerned with the development of a system which will meet Patent Office requirements -- requirements which are more stringent than those faced by any other organization.

APPENDIX A

The following people contributed to the development of HAYSTAQ through Stage Two, participating for varying lengths of time:

National Bureau of Standards

Ethel C. Marden
Catherine Lester
Robert T. Moore*
John F. Rafferty
Susan Starbird**
Alen J. Tudgay

U. S. Patent Office

Herbert R. Koller
Harold Pfeffer
H. Winston Hayward
Ernestine (Connor) Bartlett
Yvonne Harris
Dale R. Mahanand
George F. Fraction
James H. Turnipseed
Helen M. S. Sneed
Gregory E. McNeill
Harry W. Royal
Galen S. Marburg**
Raymond L. Bridge**
Ellen Isaacs**

*Summer student work only
**Summer students who worked on auxiliary implementation of HAYSTAQ which have not been put into practice at this time.

APPENDIX B will be found as the
last page of this document

APPENDIX C

PART I. ERROR CHECKING

The chemical structure data undergo a drastic transformation from their original form of structure diagrams with some descriptive text to the machine stage of perforated paper tape containing punched holes representing the hexadecimal-coded structure data. In the wide range of human and machine activity necessary to accomplish this document-to-tape transformation there are many opportunities for errors. Safeguards were built into the system where feasible, but the more machine-oriented the data become, the more difficult and tedious it is to discover errors by visual inspection.

Analysis of the structure information was carried out in an attempt to discover what sorts of errors might occur and so to provide guidelines to make a determination of correction procedures. The ordinary typing errors were caught by the mechanical comparator of the punched paper tapes; transcription errors resulting from badly written characters were sometimes caught because of a difference of opinion between the two typists, but usually were not detected before the computer processing of the raw data. To perform more elaborate and thorough-going error checks, various logical processes were required. These called for decisions among a variety of permissible options. The study of this type of error resulted in the development of the error-checking routine SWEEP.[1/] A list of the kinds of errors detected by SWEEP is contained in the "Error Dictionary" which comprises the latter part of this Appendix. In addition to detecting errors, SWEEP maintained records of the types of errors it detected and the frequency of such errors. The kinds of errors encountered in transforming the data from one representation into another were roughly divided into five categories:

1. Local language convention errors. The rules or conventions of an arbitrary language were violated, e.g., substitution of the wrong designation for a category of information.

2. Global language errors. A large structure of the material is wrong, although small arbitrary details may be correct, e.g., omission of a descriptor containing the complete definition of a functional group. One indication of this type of error is the instance when other descriptors show a connection to something that is not there.

3. Inconsistency errors. Information contained in one part of the data does not check properly with redundant information contained in another area. The "usefully redundant" information, although space-consuming, is a valuable property of data for the purpose of mechanized data checking.

4. Conflicts with natural laws. This mistake results in a situation which could not occur in nature; e.g., no halogen can occur triple-bonded to any other functional group. Such checks for impossible occurrences will occasionally reveal errors not detected by other means.

5. Informational errors. This error is not an accurate transform of the original information, and may result from a misunderstanding or a logical blunder.

1/ SWEEP and SQUASH were written by Robert T. Moore.

A knowledge of the rules of the coding language is required for detection of the first two types of errors; understanding of the symbol meanings is required for type three; understanding of chemistry is required for type four; and a second source of information about the specific structure is required to detect errors in type five. It was not deemed practicable to try to provide at this time for the more elaborate checks required for types four and five, partially because of limited storage; therefore, only the first three types of error checks were included in SWEEP.

When an error was detected, the typewriter printed out the type of error found, the identification number of the disclosure in which it occurred, the designation number of the incorrect descriptor, and the relative location of the erroneous word. In addition, SWEEP kept records of rejected disclosures including such useful information as the date the disclosure analysis and coding was completed, the date of the final (verified) punched paper tape, the number of words contained in the disclosure, the name of the analyst who encoded the disclosure, and the number of times the disclosure had been rejected previously by SWEEP for other errors.

SQUASH performed the same functions for the question data which SWEEP performed for the disclosure.

PART II HADACOR, SAND, AND SQUAD [1/]

HADACOR is a short routine used in conjunction with SWEEP to correct and reprocess those disclosures previously rejected because of errors. HADACOR makes the required corrections according to the rules (which vary with the nature of the correction) and modifies certain control instructions in the SWEEP program. This modification permits HADACOR and SWEEP to operate as a single routine so that the corrected disclosure may then be rechecked and, if no further errors are found, added to the disclosure file.

SAND stores the corrected structure data of the disclosure in the computer's storage locations allocated for them, then examines each of the data words in sequence for processing. (See Figure 1 for unprocessed data formats.) The structure data are compressed, re-arranged, and assembled in the word formats required by the chemical structure search routine, and each such encoded word is then transferred to the next available address in the storage locations allocated for the assembled word storage. (See Figure 2 for the assembled data formats.) Descriptors within each of the three sections are then sorted and transferred, in sequence, to that section of the storage allocated for the sorted and assembled data storage. During the assembly and sorting operations, additional information required by the search routine for housekeeping operations is computed. In addition, the arbitrarily assigned numbers of the 3C descriptors [2/] are replaced by reference symbols designating the position of the first word of each such descriptor relative to the beginning of the 3C data. The symbols thus serve as location finders in memory for the descriptors and provide a relative address which is used in the structure search routine for each such descriptor. These reference symbols are also substituted for the previously assigned arbitrary numbers in the "designation number definition" ("DN Def") and "designation number connection" ("Dn Conn") fields of the 3B and 3C descriptors, respectively. (See data formats at the end of this Appendix.)

1/ These three routines were written by Catherine E. Lester.

2/ Each such 3C descriptor is made up of several words containing descriptive information for one of the functional groups in the structure.

All of the 3A terms are sorted and sequenced in ascending order, and a flag word is inserted after the last 3A term to denote the end of the section. The 3C words are not sorted except for Markush groups; each Markush group is internally sorted and sequenced, with the members of the group arranged in numerically ascending order by designation number. A flag is also used to denote the end of the Markush group. The 3B data are sorted at three levels (except for ring descriptors, which are only sorted once). The first-level sort is an intra-descriptor ordering of the reference symbols in the "DN Def" fields. They are sorted in ascending order by the numerical value of the substantive codes of the 3C descriptors (which are the definitions of the 3B generic terms), and the sort in each case here has to transfer to a consideration of the 3C data codes before the 3B sorts can be completed. In addition, the old designation numbers in the "DN Def" fields are replaced by their new reference symbols (relative addresses). After completion of the first-level sort, the 3B descriptors are sorted and sequenced at the second level by the numerical values of their substantive codes. If more than one with the same substantive code is found, these substantive codes are sorted by referring to the 3C descriptors defining them; such 3C descriptors are sorted by their substantive codes, and the smallest such substantive code determines which of the like substantive codes in the definition fields at the 3B level will be first listed. There may be many multiples of like terms, and the sorts can sometimes become rather complicated.

When all data have been encoded, compressed, assembled, sorted and stored, there is computed some housekeeping information with respect to storage requirements; read-in instructions for the structure search routine are also supplied.

The detailed processing of the data for each type of word format is not described here, as it is probably not of interest to the casual reader. The processing of the 3B data in particular is quite complicated. In-house documentation exists for a complete description of the SAND processing.

SQUAD is the computer program for compressing, assembling, and re-organizing the corrected question structure data. The question data have many information fields common to those of the disclosure data, but they have additional information fields in order to provide greater flexibility for the questioner. (See Figure 2.) SQUAD is a modified version of SAND, but it also incorporates instructions for compressing and assembling the types of information not found in the disclosure data.

Records required for an orderly handling of the data processing included (1) the number of times a particular disclosure was returned to the chemist-analyst for correction; (2) the stage of processing of a particular disclosure at any point in time, i.e., whether on punched paper tape only, whether processed by SWEEP once, twice, or more, whether processed by SAND; (3) the level of completion of data preparation for any particular document, i.e., whether some disclosures in the document are absent because of corrections to be made as a result of SWEEP processing; (4) the number of completed (document) entries on magnetic tape; (5) the number of disclosures rejected because they were of a size too large to be accommodated in the available memory storage of SEAC; (6) total number of disclosures in the file at any point in time, as well as the average number of disclosures per document, number of words per disclosure, and the smallest and largest disclosures in the file. Similar records were maintained for question data processing.

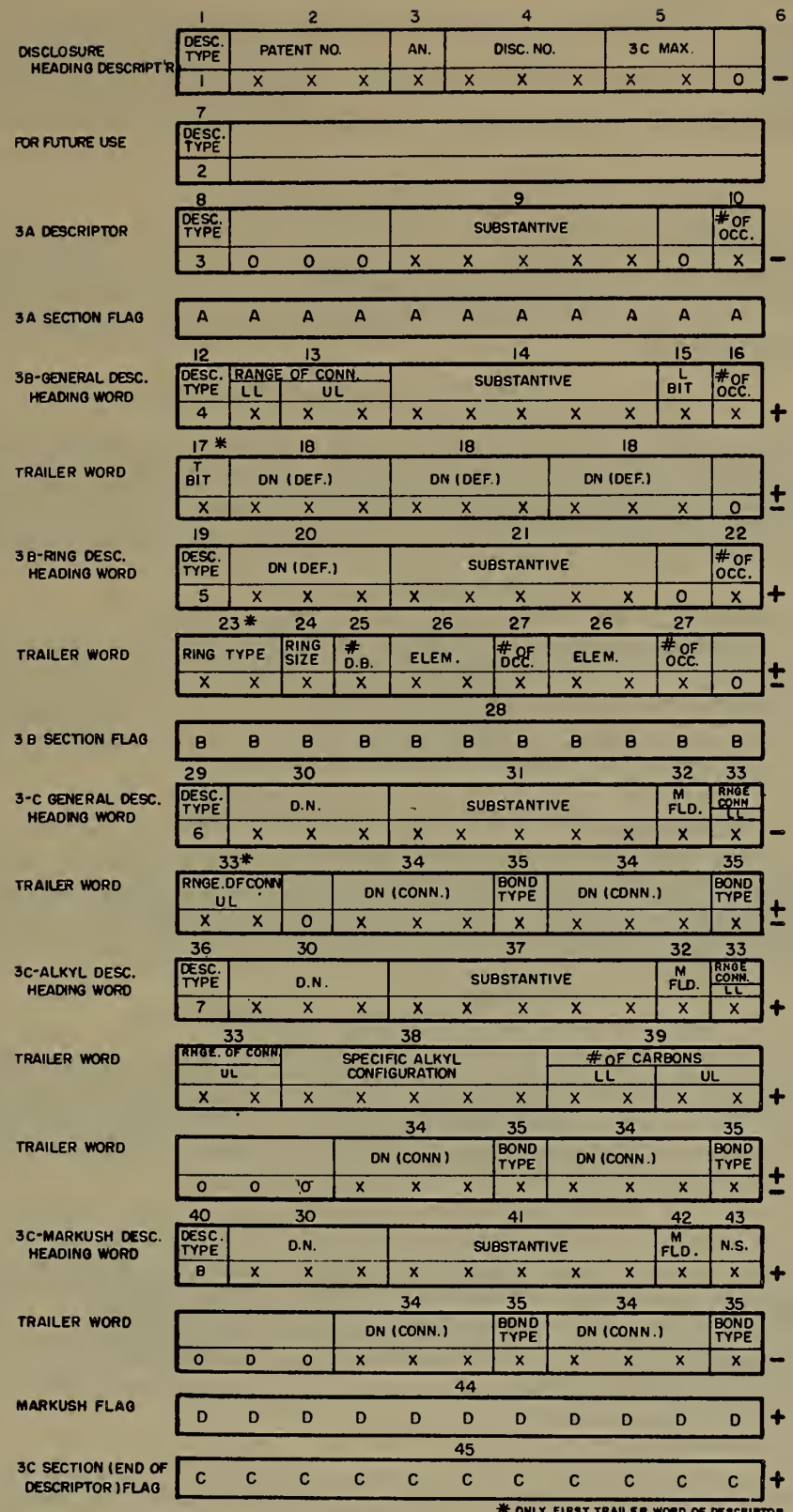


FIGURE 1. FORMAT OF RAW DATA DISCLOSURE WORDS

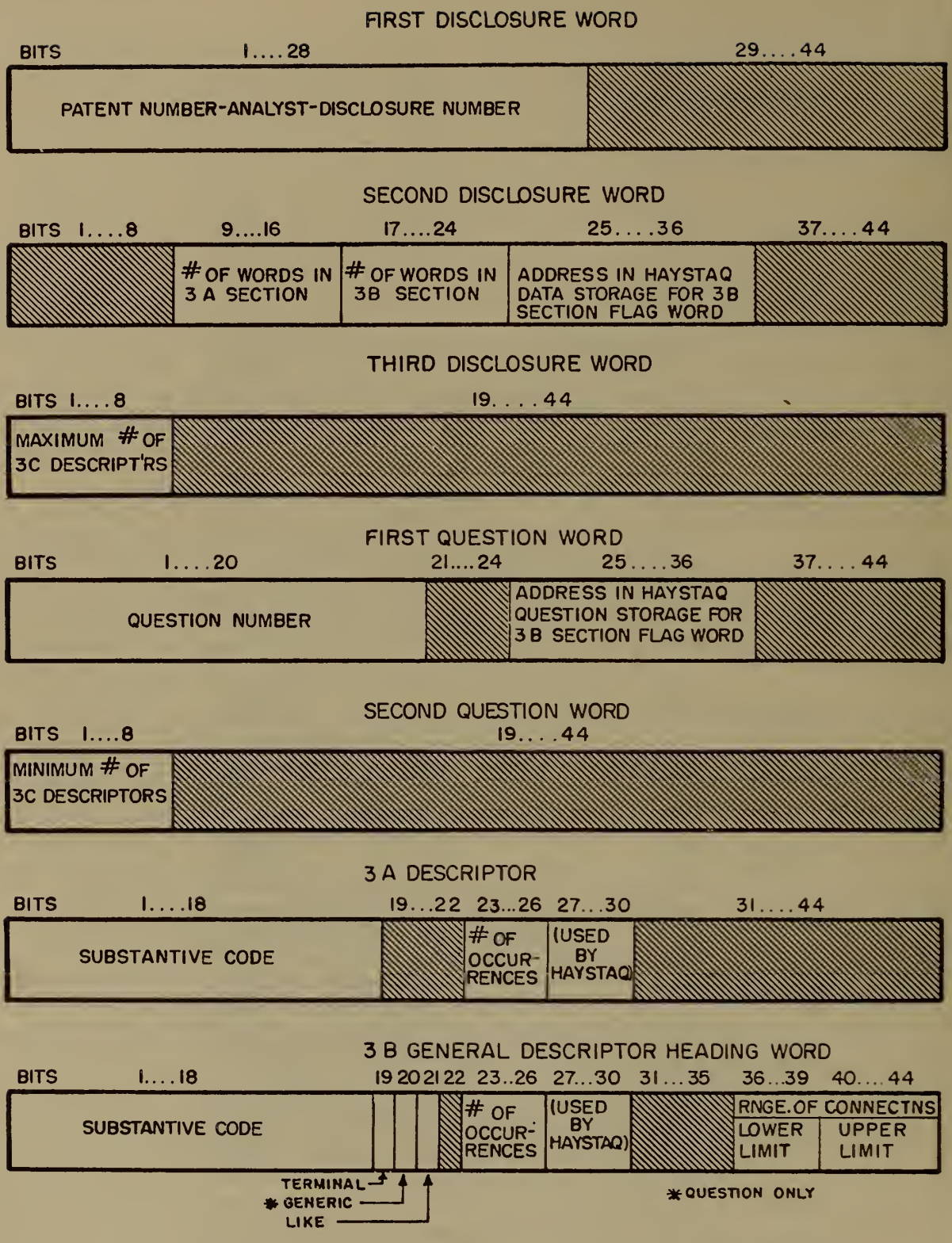
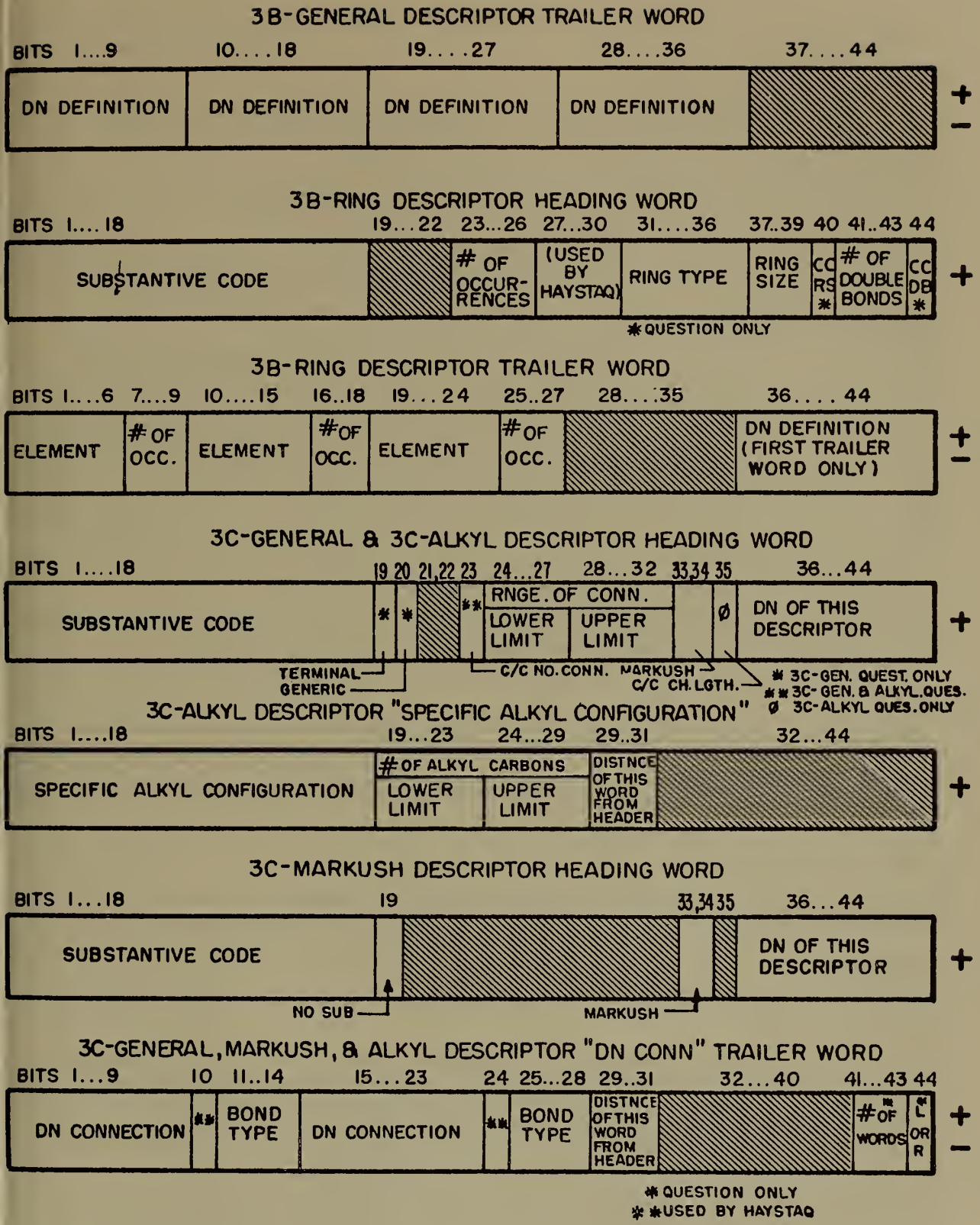


FIGURE 2. FORMATS OF "SAND-ED" DISCLOSURE AND QUESTION DATA WORDS

(PAGE 1 OF 2)



(Figure 2. (continued)

Disclosure Heading Descriptor

Field No. 1. Descriptor Type 1 (DT 1). A label that indicates that the descriptor carrying it is the particular descriptor which identifies a disclosure and provides general information about it (i.e., the disclosure heading descriptor).

Field No. 2. Patent Number. A number in this disclosure heading descriptor which tells in which patent the disclosure occurred.

Field No. 3. An. (Analyst). This indicates which of the Patent Office staff of analysts analyzed the patent, extracted the disclosure, and encoded it as it appears on tape.

Field No. 4. Disclosure Number. An identifying number specifying which of the disclosures in the given patent is described.

Field No. 5. 3C Maximum. The maximum number of Functional Groups occurring in the disclosure. This number is used for screening, since a request for a larger structure cannot possibly be satisfied by the disclosure. Due to the possibility of Markush genera, if "Markush groups" occur in the disclosure there will be several alternative structures represented by the disclosure. This number corresponds to the largest structure that can be made up from any grouping of members from the Markush groups. (One member of any Markush group could well be "No Substituent (NS)", which would have the effect of decreasing by one the number of potential functional groups in the structure containing that Markush group.)

Field No. 6. Minus Sign. Always signifies the end of a descriptor.

Field No. 7. Descriptor Type 2 (DT 2). Reserved for future use when documents othe than patents are coded. DT 1 will be patent data, DT 2 non-patent data.

3A Descriptor

Field No. 8. Descriptor Type 3. A label indicating that the descriptor to which it belongs is 3A data.

Field No. 9. Substantive (Type 3). This number is the name of some large (multi-group) substructure occurring in the disclosure.

Field No. 10. Number of Occurrences (Type 3). Specifies the number of times the large substructure indicated in Field No. 9 occurs in this disclosure.

Field No. 11. A Flag (All A's). An indicator word that signals the end of the 3A data for this particular structure.

3B Descriptor

Field No. 12. Descriptor Type 4. A label that denotes a descriptor of the 3B generic data type.

Field No. 13. Rg. Conn. (Range of Connections) UL (upper limit) and LL (lower limit). A generic term may have just functional groups, or sometimes entire substructures, as embodiments. In each case the embodiment, considered as a unit,is connected to other functional groups. Markush groups with NS members <43> (unrepresented hydrogens) permit a variation in the number of possible connections, hence the variability of the number of connections and the necessity for UL and LL numbers.

Field No. 14. Substantive (Type 4). Contains the name of a genus (halogen, amide, ester) which has specific embodiments described in the 3C data in this disclosure.

Field No. 15. L Bit (Like Bit). If 1, indicates that the descriptor in which it occurs has the same substantive field as the preceding one. If "0", the two descriptors have different substantives.

Field No. 16. # Occ. (Number of Occurrences). Generally 1. If a given substantive occurs in more than one descriptor, the # Occ. of the first contains the total number of occurrences, and all the following descriptors with this substantive have a "0" in # Occ.

Field No. 17. T Bit (Terminal Bit). Indicates whether some one of the embodiments of a genus is terminal. A terminal group is one which is connected to only one other functional group in the structure (the end of a line, in other words).

Field No. 18. DN (Def). These fields contain the designation numbers (arbitrary designations) of the descriptor type 6 or 7 functional groups that occur in specific embodiments of the genus, and which are described by the 3C data.

Field No. 19. Descriptor Type 5. Indicates that its descriptor is a 3B ring descriptor, one which describes in detail the composition of a particular ring structure.

Field No. 20. DN (Def). Gives the designation number (arbitrary designation) of the ring (functional group) occuring in 3C, for which this descriptor gives certain details.

Field No. 21. Substantive. The same symbol for all rings, means "ring".

Field No. 22. # Occ. Indicates the number of rings in the 3B table and is used only in the first ring listed.

Field No. 23. Ring Type. Provides for general descriptions, such as carbocyclic or heterocyclic or for specitic descriptions such as aromatic.

Field No. 24. Ring Size. Indicates the number of atoms in the ring being described.

Field No. 25. # D.B. (Double Bonds). Indicates the number of double bonds in the ring.

Field No. 26. Elem. (Element). Indicates one of the elements present in the ring (tells what type it is).

Field No. 27. # Occ. (Element). Indicates how many atoms of the particular element listed in Field No. 26 are present in the ring.

Field No. 28. B Flag (All B's). Indicates the end of the 3B table.

3C Descriptor

Field No. 29. Descriptor Type 6. Indicates a general 3C descriptor, which is used to describe all functional groups except alkyls and Markush groups.

Field No. 30. DN (Designation Number). An arbitrary number , one of which designates each separate functional group in the structure. A separate designation number is required for each such group, since a given type of functional group may occur more than once in a disclosure).

Field No. 31. Substantive. The name of the type of functional group described in the descriptor (Keto, phenyl...).

Field No. 32. M Fld. (Markush Field). Applies only in descriptors for functional groups contained in Markush groups. It indicates how many connections occur between the given group and other groups not in the Markush group .

Field No. 33. RG Conn UL and LL. Representation of the range in the allowed number of other functional groups connected to the given functional groups. As explained in <13>, Markush NS <43> possibilities permit the variation between LL and UL.

Field No. 34. DN Conn (Connection Designation Number). Indication of a connection

from the functional group described in this descriptor to the one with the designation number given in this field.

Field No. 35. Bond Type. Specifies whether the bond given by the DN Conn recognized in Field No. 34 is single, double, triple, fused (applies-to rings), doubly fused, or spiro.

Field No. 36. Descriptor Type 7. Indicates a descriptor for an alkyl group. Descriptors of this type differ only slightly from Type 6.

Field No. 37. Substantive. All alkyls have the same fixed substantive name (alkyl) even though this is a generic term.

Field No. 38. Specific Alkyl Configuration. If a specific alkyl is given in the disclosure, this code identifies the alkyl and indicates which of its isomers was given.

Field No. 39. # of Carbons. In case a class of alkyls is allowed, rather than some specific alkyl, the UL (Upper Limit) and LL (Lower Limit) may be different. They indicate the allowed variation in number of carbons, otherwise UL = LL.

Field No. 40. Descriptor Type 8. Indicates a "Markush Heading descriptor". In the case of a Markush genus, all "outside" connections to the Markush group are shown to this descriptor, which is then connected back to these outside groups (only one or two are allowed). This descriptor otherwise differs only a little from Types 6 and 7.

Field No. 41. Substantive. All Markush descriptors have the same fixed substantive name (Markush).

Field No. 42. M. Field. Serves the function of indicating whether the Markush group has one or two outside connections. Replaces RG CONN information.

Field No. 43. NS Bit (No Substituent). Indicates whether or not the Markush genus contains a hydrogen, which is regarded as "not there".

(REMARK: All members of a Markush genus are described in terms of their functional group or groups, immediately following the Type 8 descriptor representing the genus.)

Field No. 44 D (Markush) Flag (All D's). Indicates the end of a Markush group.

Field No. 45. C (End) Flag (All C's). Indicates the end of the disclosure.

Symbol	Cause
EEE EEE EEE EE	The heading word of this disclosure does not have a descriptor type 1. This means (in most cases) that the read-in is out of synchronization, either by a few characters, or a few words. This exit thus concerns primarily the computer operator. Pickup of stray bits or other errors can cause a "legitimate" error of this type, however.
EO	The number of words in this disclosure (as given in the word count) exceeds the storage space available for it in SEAC (168_{16} or 360_{10}). This may merely indicate out-of-phase read-in, due either to machine failure or to an omitted carriage return at the beginning of the disclosure. On the other hand, it may be a spurious error due to too small a word count in the "lost" disclosure, which will have caused only part of the disclosure to be read in. This means a data word will be interpreted as the word containing the date and word count.
E1	The location of the end flag (CC...C) is inconsistent with the word count. Read-in terminates at end flag or when specified number of words are read in, whichever occurs first. Note here that if word count is low by 1 or 2, the next disclosure will not be correctly placed for read-in.
E2	The heading word is not negative, or some odd rearrangement of words has occurred.
E3	Now defunct; will never be encountered.
E4	The 3A descriptor printed out for examination is not of descriptor type 3. This may indicate absence or mutilation of the 3A end flag.
E5	There is some sort of number in the designation number field of this type 3 descriptor. It may be a mislabeled and misplaced 3B or 3C descriptor (the descriptor is printed out for examination).
E6	The number of occurrences of this 3A substantive term has been omitted (i.e., - is given a zero). The descriptor is printed out.

Symbol	Cause
E7	This 3A (type 3) descriptor is not negative.
E8	There is no 3A end flag (a word of all A's) anywhere in the disclosure (omission of 3A end flag will more often actuate E4). It is desirable to check for dropped bits in the flag word.
E9	There is no 3B end flag (all B's) in this disclosure. Check for dropped bits in the flag word.
E10	There is a minus sign in the wrong place somewhere in the last few words of the disclosure. (This error print arises in trying to identify the last designation number in 3C).
E11	The total number of inter-group connections in the structure is odd. Since, for groups A and B, A is shown connected to B and vice versa, each connection occurs twice. If A is shown connected to B twice (B's designation number appears twice in A's connection field) the indicator will be actuated. This error occurs in 3C data only.
E12	The substantive word of a 3C descriptor is given as negative (this word is printed out).
E13	The descriptor type of a 3C word is not between 6 and 8. This may occur if there is a minus sign in the wrong place in the 3C data, or if a flag word (Markush most likely) has dropped a bit. The offending word is printed out.
E14	The designation number of two successive 3C descriptors differs by something other than 1. Thus either a descriptor has been omitted, or some descriptors are out of numerical order. It is conceivable that this error print might be actuated if a word which is <u>not</u> the substantive word of a descriptor happens to occur preceded by negative word and with a 6, 7, or 8 in the descriptor type field (this is <u>very</u> unlikely). The two designation numbers are printed out.
E15	The lower limit to the range of connections in this descriptor is greater than the upper limit. The designation number of the descriptor is printed out.

Symbol	Cause
E16	The upper limit of connections in this descriptor is shown as zero, but this is not shown as a one-group structure. (3C max > 1 in the disclosure heading word). The designation number of the descriptor is printed out.
E17	The upper limit of the range of connections is non-zero, but 3C max was shown as 1. The designation number of the descriptor is printed out.
E18	Inoperative, will never occur in practice.
E19	A Markush heading descriptor (descriptor type 8) occurs in a one-group structure (3C max = 1). The designation number of this group is printed out.
E20	A Markush heading descriptor (descriptor type 8) occurs within a Markush group. The designation number of this descriptor is printed out.
E21	The substantive field of an Alkyl (type 7) descriptor is not "3CFFF" as it should be. The designation number of this descriptor is printed out.
E22	The upper limit to the number of carbons in an alkyl descriptor is less than the lower limit. The designation number of this descriptor is printed out.
E23	The specific alkyl configuration field is non-zero, yet the upper limit of the number of carbons is not equal to the lower. The designation number of the descriptor is printed out.
E24	The alkyl configuration number is less than the smallest possible for the number of carbons given. The designation number is printed out.
E25	The alkyl configuration number is larger than the largest possible for the number of carbons given. The designation number is printed out.
E26	The Markush (M) field is non-zero in a descriptor which does not lie within a Markush group (A Markush flag may be misplaced). The designation number of the descriptor is printed out.
E27	The connection field of a descriptor in a one-group structure (3C max = 1) is non-zero. The designation number of this descriptor is printed out.

Symbol	Cause
E28	The connection word is positive in a descriptor in a one group structure (3C max=1). The designation number of this descriptor is printed out.
E29	The word following the single group of a one-group structure is not an end flag. This check will catch most such errors, the others will be caught by E18. The printout is irrelevant.
E30	A group is shown as connected to itself (contains its own designation number in its connection field). The designation number is printed out.
E31	A group is shown connected to another group with a designation number greater than that of the last group in 3C. This may indicate spurious bits or that the last entries in 3C are out of numerical order. The single word printed out contains the designation number of the descriptor in the DN field, and the incorrect designation number in "integer position" (the right end of the word).
E32	The group whose descriptor is being examined is recorded as connected to a group whose designation number is within the proper range for 3C (i.e., less than the last designation number), but no descriptor with that designation number appears in 3C Both the designation numbers are printed out: that of the origin group in the DN field, and that of the missing group to which it is supposedly connected in integer position (at the right hand end of the single word).
E33	In checking the connection between group A and group B, Group B has been located (that is, a word with the correct DN field has been found) but for some reason, the descriptor type of this word is not 6, 7, or 8. This probably indicates stray bits, either in the DT field of the correct descriptor, or in the DN field of some word other than a substantive word of some other descriptor. The designation number of A is in the DN field and that of B in integer position in the word printed out.
E34	Group A is shown connected to group B, and vice versa, but the bond types for this bond do not agree. The designation number of A is in the DN field, that of B in integer position in the print out. Note that

Symbol	Cause
E34 (Continued)	this check does not cross Markush group boundaries into the group, but the bond types of groups connected to the Markush group must agree with the corresponding bond types in the Markush heading (type 8) descriptor itself.
E35	The last "connection word" of a descriptor is positive. This either means that the given upper limit to the range of connections is too small, or that a minus sign has been left out. The designation number of this descriptor is printed out.
E36	The last "connection word" of a descriptor contains bits in a connection DN field that should be zero if the upper limit of the range of connections is correct. The designation number of this descriptor is printed out.
E37	The upper limit to the range of connections of a descriptor is so large that supposed "connection words" would have to extend beyond the end of the disclosure (this will occur very rarely). The designation number of this descriptor is printed out.
E38	A connection word in this descriptor is negative, but the upper limit to the range of connections indicates there should be at least one more word of connections. The designation number of the descriptor is printed out.
E39	Group A is shown connected to group B, but the designation number of group A does not appear in the connection field of group B (the range of connections, upper limit, of B may be in error). The designation number of group A is in the DN field, that of B in integer position in the printout.
E40	Group A is shown connected to group B, but a negative "back connection" word in the connection field of B has been encountered (a) before A's designation number has been found and (b) before all possible connections have been examined. This partially duplicates E38 but has a different logical function in the computer program. The designation number of A is given in the DN field, that of B in integer position in the printout.
E41	Inoperative, cannot occur in practice.
E42	The Markush heading descriptor (type 8)

Symbol	Cause
E42 (Continued)	has the wrong substantive field (not "3FF FF"). The designation number of this descriptor is printed out.
E43	The M field is zero in a Markush heading (type 8) descriptor. This exit will probably never occur. The designation number of this descriptor is printed out.
E44	The M field is not 3 in a Markush heading (type 8) descriptor, yet two connections are shown in the connection field. The designation number of this descriptor is printed out.
E45	The M field is not 1 for a Markush heading (type 8) descriptor which has only one connection shown in the connection field. The designation number of the descriptor is printed out.
E46	One of the connection fields of this descriptor is blank, although the range of connections and the location of the minus sign indicate that there are more connections. The designation number of this descriptor is printed out.
E47	There is no Markush flag (all D's) between a Markush heading (type 8) descriptor and the end of the disclosure. The printout is irrelevant.
E48	A connection shown for a group within a Markush group is neither to one of the other groups within the Markush group, nor to one of the groups shown as connected to the Markush heading descriptor. The designation number of the descriptor with the bad connection field is printed out.
E49	A group within a Markush group is shown as connected to the Markush heading descriptor (it should be shown as connected to some other group, and the designation number of that group should also appear in the connection field of the Markush heading descriptor). The designation number of the descriptor is printed out.
E50	The M field of a Markush heading descriptor to which group A is shown as connected is not 1 or 3. (This check duplicates E43-E45 but serves a different logical function in the checking program.) The designation number of group A is in the DN field and that of the faulty type 8 descriptor in integer position in the printout.

Symbol	Cause
E51	A Markush heading (type 8) descriptor occurs, followed immediately by a Markush flag, hence the members of the Markush group have either been omitted or placed after the flag. The designation number of the Markush heading descriptor is printed out.
E52	The first word of a 3B (type 4 or 5) descriptor is negative (this may be a 3A descriptor with the wrong DT field and in 3B). The first word of this descriptor is printed out.
E53	A 3B descriptor has a descriptor type different from 4 and 5. This indicator may be actuated if some DN (Def) or element word which should be positive is negative. In that case, the next DN (Def) or element word will be interpreted as a substantive word when in fact it is not. The (supposed) substantive word is printed out.
E54	In a type 4 (General) 3B word, the lower limit to the range of connections is greater than the upper limit (possibly this descriptor is actually a mislabeled ring descriptor). The substantive word of the descriptor is printed out.
E55	There is an error in the upper limit of the ranges of connections in this type 4 descriptor. Either it is zero for a multi-group structure, or non-zero for a one group structure. The substantive word of the descriptor is printed out.
E56	The L bit of this descriptor is a 1 but should not be a 1 if the "number of occurrence" information in preceding descriptors is correct (this may simply mean that some "number of occurrence" value is too small. Substantives have not been checked.) The substantive word of this descriptor is printed out.
E57	According to L bit and "number of occurrences" information, this descriptor should have the same substantive as its predecessor, but it does not. The substantive word is printed out.
E58	The "number of occurrences" field of this descriptor is 0 and so is its L bit (number of occurrences may be blank only when L bit is a 1). The substantive word of the descriptor is printed out.

Symbol	Cause
E59	The substantive field of this descriptor is the same as that of the last, yet its L bit is 0 and the number of occurrences given in the previous descriptor was 1. The substantive word is printed out.
E60	The T bit in this descriptor > 1: this occurs in a descriptor word which is supposedly a "DN (Def)" word. In most cases this indicates that a minus sign has been left off the last "DN (Def)" word, causing the program to interpret the next descriptor's substantive word as a "DN (Def)" word (reading the descriptor type as a T bit). This is the only check we have on missing signs in type 4 descriptors (note that spurious bits in a T field may also actuate this exit, however). The substantive word of this descriptor is printed out.
E61	The T bit of this descriptor is inconsistent with the "range of connections" data. If $LL \leq 1 \leq UL$, the T bit must be a one, otherwise it must be a zero. The substantive word of this descriptor is printed out.
E62	A type 5 (Ring) descriptor does not have the proper ring substantive ("2BF59"). The substantive word of this descriptor is printed out.
E63	A type 5 (Ring) descriptor has a "DN (Def)" field outside the 3C range (either zero or too large). The substantive word of this descriptor is printed out.
E64	The ring size is not equal to the number of occurrences of the first element in a homocyclic (20 series)ring (type 5) descriptor. The substantive word of this descriptor is printed out.
E65	The ring type of a ring (type 5) descriptor is greater than the maximum possible value, 28 (hexadecimal). The substantive word of this descriptor is printed out.
E66	A non-carbocyclic (ring type 28) ring (type 5) descriptor has the carbon symbol (1) in its element field. The substantive word of this descriptor is printed out.
E67	A carbocyclic (ring type 24, 25, or 26) ring (type 5) descriptor contains a symbol

Symbol	Cause
E67 (Continued)	other than carbon (1) in its element field. The substantive word of the descriptor is printed out.
E68	A non-carbocyclic (ring type 28) ring (type 5) descriptor has an illegitimate code in its element field (i.e., a code greater than D (hexadecimal)). The substantive word of the descriptor is printed out.
E69	A carbocyclic ring (type 5) descriptor has 21-23 or 27 in its ring type field. The substantive word of this descriptor is printed out.
E70	An aromatic carbocyclic (ring type 25) ring (type 5) descriptor is shown with a number of carbons other than six. The substantive word of this descriptor is printed out.
E71	The terminal word of a homocyclic (20 series) ring (type 5) descriptor is positive. The substantive word of this descriptor is printed out.
E72	Element and number of occurrence fields, in a homocyclic (20 series) ring (type 5) descriptor, which should be blank are not. The substantive word of this descriptor is printed out.
E73	Inoperative.
E74	The elements are not in the correct (numerical) order in a heterocyclic (10 series) ring (Type 5) descriptor. If this is a non-carbon (ring type 18) ring, the presence of carbon (symbol 1) in the element field actuates this error. The substantive word of this descriptor is printed out.
E75	The last element word (as determined by comparison of ring size and number of occurrence information) of a heterocyclic (10 series) ring (Type 5) descriptor is positive. The substantive word of this descriptor is printed out.
E76	The sum of the number of occurrences of the different elements in a heterocyclic (10 series) ring (type 5) descriptor is greater than the given ring size. The substantive of this descriptor is printed out.
E77	An element word of a heterocyclic (10 series) ring (type 5) descriptor is negative before all elements have been listed (according to a comparison of ring size

Symbol	Cause
E77 (Continued)	and number of occurrences information). The substantive word of this descriptor is printed out.
E78	The last hexadecimal digit of a DN (Def) word of a general (type 4) descriptor is non-zero (it must always be 0, otherwise it causes trouble in the data assembly program). The substantive word of this descriptor is printed out.
E79	A DN (Def) in a general (type 4) descriptor is outside the range of designation numbers occurring in 3C. The substantive word of this descriptor is printed out.
E80	The number of occurrences given in the first type 5 descriptor is not equal to the number of type 5 (ring) descriptors in 3B. The printout is irrelevant.
E81	The ring type of a ring (type 5) descriptor is within the heterocyclic range (10-18) but is not one of the proper values (10, 14, or 18). The substantive word of this descriptor is printed out.
E82	The substantive field of a 3B (type 4) descriptor contains a number outside the range "2B001-2BF58". The substantive word of this descriptor is printed out.
E83	The substantive field of a 3A (type 3) descriptor contains a number outside the 3A range "10647-1AFFF". The substantive word of this descriptor is printed out.
E84	The substantive field of a 3C (type 6 only) descriptor contains a number outside the 3C range "300001-3CFFF". The designation number of this descriptor is printed out.
E85	There is a type 4 descriptor mixed in with the type 5 descriptors (or following them) in 3B. The substantive word of the misplaced descriptor is printed out.
E86	There are more than two words of DN (Def)'s in a 3B type 4 descriptor. Thus it cannot be accommodated by the SAND routine. The substantive word of this descriptor is printed out.
E87	The 3C descriptor defined by a 3B type 5 ring descriptor (that is, the 3C descriptor with the same DN as given in the DN (Def) field of the type 5 word) has a substantive field outside the 3C ring range ("30001-

Symbol	Cause
E87 (Continued)	30646"). The substantive word of the 3B descriptor is printed out.
E88	There are too many words in this 3C (type 6 or 7) descriptor. For both types, up to 8 words in all is permissible (this gives up to and including 7 connection words for type 6, and 6 connection words for type 7, which has the extra alkyl configuration word). The designation number of the descriptor is printed out.

The technical contributions described in the substantive portions of this report result from the combined efforts of personnel from the National Bureau of Standards and the U. S. Patent Office. In particular, the author wishes to acknowledge the contributions made to the system design by Mr. Harold Pfeffer and Mr. Herbert R. Koller, and the subsequent assistance by Mr. Koller in the program debugging operations and the computer runs. Mr. Koller's paintaking analysis of the discrepancies between the expected answers and actual answers from the machine searches formed the basis for the analysis of the results of computer runs which is presented in Chapter VIII.

The careful work of the chemist-analysts and their supervisors assisted greatly in attaining the objectives of the HAYSTAQ program. The chemists who analyzed the patents at the U. S. Patent Office were recruited and trained by Mr. Pfeffer. In the production stage of their work, they worked primarily under the supervision of Mr. H. Winston Hayward, also of the U. S. Patent Office. Their names are listed in Appendix A.

Accumulation of the file of encoded patents which was used to make the searches was made possible only because of the many hours of patient effort contributed by Miss Catherine E. Lester, Mr. John F. Rafferty, and Mr. Alan J. Tudgay. Miss Lester supervised the mechanized processing of the file information, from establishing procedures for quality control of the initial punched paper tape output to the final assembly of checked, encoded data onto magnetic tape. Intermediate stages of the operation included machine checking of the original encoded data, the return of incorrect information to the chemist-analysts for correction, reprocessing of the corrected information, and assembly into final form of the data. Mr. Rafferty helped to streamline procedures for more efficient data handling, prepared some of the input routines for the computer programs, and rendered valuable assistance in all debugging operations on SEAC. He and Mr. Tudgay maintained and operated SEAC in making the long, tedious computer runs required to accumulate the mechanized file and to make the chemical structure searches which followed.

Acknowledgment is hereby made to the following individuals for their careful reading of this report, and for the helpful suggestions which they made to improve its content.

Mr. Ezra Glaser, Assistant Commissioner of Patents, U. S. Patent Office
Mr. Samuel N. Alexander, Chief, Information Technology Division, National Bureau of Standards
Mr. Herbert R. Koller, U. S. Patent Office
Dr. Stephen J. Tauber, National Bureau of Standards
Miss Mary E. Stevens, National Bureau of Standards
Miss Margaret R. Fox, National Bureau of Standards

REFERENCES

1. U. S. Department of Commerce Advisory Committee on Applications of Machines to Patent Office Operations, V. Bush, Chairman, Report to the Secretary of Commerce (Department of Commerce, Washington, D.C., December 22, 1954).

2. B. E. Lanham, J. Leibowitz, H. R. Koller, and H. Pfeffer, "Organization of Chemical Disclosures for Mechanized Retrieval", paper presented at American Chemical Society, 131st Meeting, Miami, Florida, April 8, 1957. Also Patent Office R & D Report No. 5, June 14, 1957.

3. H. Pfeffer, H. R. Koller, and E. C. Marden, Am. Documentation, 10, 20 (1959).

4. H. Pfeffer and R. Swanson, "Parameters for an Information Retrieval System for Chemical Processes", Patent Office R & D Report No. 20, April 1961.

5. Ethel C. Marden and Herbert R. Koller, "Present Status of Project HAYSTAQ", pp. 163-177 in Information Retrieval Among Examining Patent Offices (based upon papers presented at the Third Annual Meeting of the Committee for International Cooperation in Information Retrieval Among Examining Patent Offices (ICIREPAT), at the Patent Office of the Federal Republic of Austria), H. Pfeffer, Editor, Baltimore, Md., Spartan Books, Inc., 1964; MacMillan and Co. Ltd., 347 pages.

6. S. M. Newman, "Problems in Mechanizing the Search in Examining Patent Applications", Patent Office Research and Development Reports, No. 3 (Department of Commerce, Washington, D.C., 1956).

7. B. E. Lanham, J. Leibowitz, and H. R. Koller, "Advances in Mechanization of Patent Searching--Chemical Field", Patent Office Research and Development Reports. No. 2 (Department of Commerce, Washington, D.C., 1956).

8. L. C. Ray and R. A. Kirsch, Science, 126, 814 (1957).

9. H. R. Koller, E. Marden, and H. Pfeffer, "The HAYSTAQ System: Past, Present, and Future" in Proceedings of the International Conference on Scientific Information, Washington, D.C., November 16-21, 1958" (National Academy of Sciences -- National Research Council, Washington, D.C., 1959), p. 1143 ff.

10. A. Opler and T. R. Norton, "New Speed to Structural Searches", Chem. and Eng. News, 34, 2812 (1956).

11. H. W. Hayward, "A New Sequential Enumeration and Line Formula Notation System for Organic Compounds", Patent Office Research and Development Reports, No. 21, (Department of Commerce, Washington, D.C., 1961).

12. Hayward, H. W. "A Second Look at Chemical Notation Systems", Proceedings, entitled Automation and Scientific Communication, of the 26th Annual Meeting of the American Documentation Institute, October 6-11, 1963, Chicago, Ill.

13. S. J. Tauber and H. W. Hayward, "Chemical Structure as Information --Representations, Calculations, Transformations", paper presented at a symposium on Technical Preconditions for Information Retrieval, conducted by the Special Interest Group on Information Retrieval of the ACM, Moore School of Engineering, April 24-25, 1964, Spartan Books, Baltimore, Maryland.

14. N. S. Prywes, H. J. Gray, et al, "The Multi-List System", Technical Report No. 1, Volumes I and II, the Moore School of Electrical Engineering, University of Pennsylvania, November 30, 1961.

15. J. C. Shaw, A. Newell, H. A. Simon, and T. O. Ellis, "A Command Structure for Complex Information Processing" in "Proceedings of the Western Joint Computer Conference, Los Angeles, Calif., May 6-8, 1958"(American Institute of Electrical Engineers, New York, New York, 1959), p. 119 ff.

16. E. H. Sussenguth, "Structure Matching in Information Processing", Scientific Report No. ISR-6 to the National Science Foundation (The Computation Laboratory, Harvard University, Cambridge, Mass., 1964).

17. G. M. Dyson, W. E. Cossum, M. F. Lynch, and H. L. Morgan, Inform. Stor. Retr., 1, 69 (1963).

18. Arthur D. Little, Inc., "Feasibility of Establishing an Integrated Agency-Wide Scientific Information System", Report to Food and Drug Administration, July 1964.

19. E. Meyer, "Encoding of Organic-Structural Formulas and Reactions by Machine" in "American Documentation Institute, 26th Annual Meeting, Chicago, Ill., October 1963, Short Papers", p. 131.

20. E. C. Marden and H. R. Koller, "A Survey of Computer Programs for Chemical Information Searching", National Bureau of Standards Technical Note 85 (Department of Commerce, Washington, D.C., 1961).

21. I. M. Hunsberger, D. E. H. Frear, R. E. Harmon, and E. G. Smith, "Survey of Chemical Notation Systems" (National Academy of Sciences -- National Research Council, Washington, D.C., 1964).

22. H. Winston Hayward, Helen M. S. Sneed, James H. Turnipseed, and Stephen J. Tauber, "Some Experience with the Hayward Linear Notation System", presented in part at the 147th National Meeting of the American Chemical Society, Philadelphia, April 5-10, 1964. Abstracts of Papers, 147th Meeting, American Chemical Society, April 5-10, 1964.

23. P. M. McDonnell and R. F. Pasternack, "A Line-Formula Notation System for Coordination Compounds", Journal of Chemical Documentation, January 1965

24. R. F. Pasternack and P. M. McDonnell, "Designation of Ligand Positions in Coordination Complexes", Inorganic Chemistry (in press).

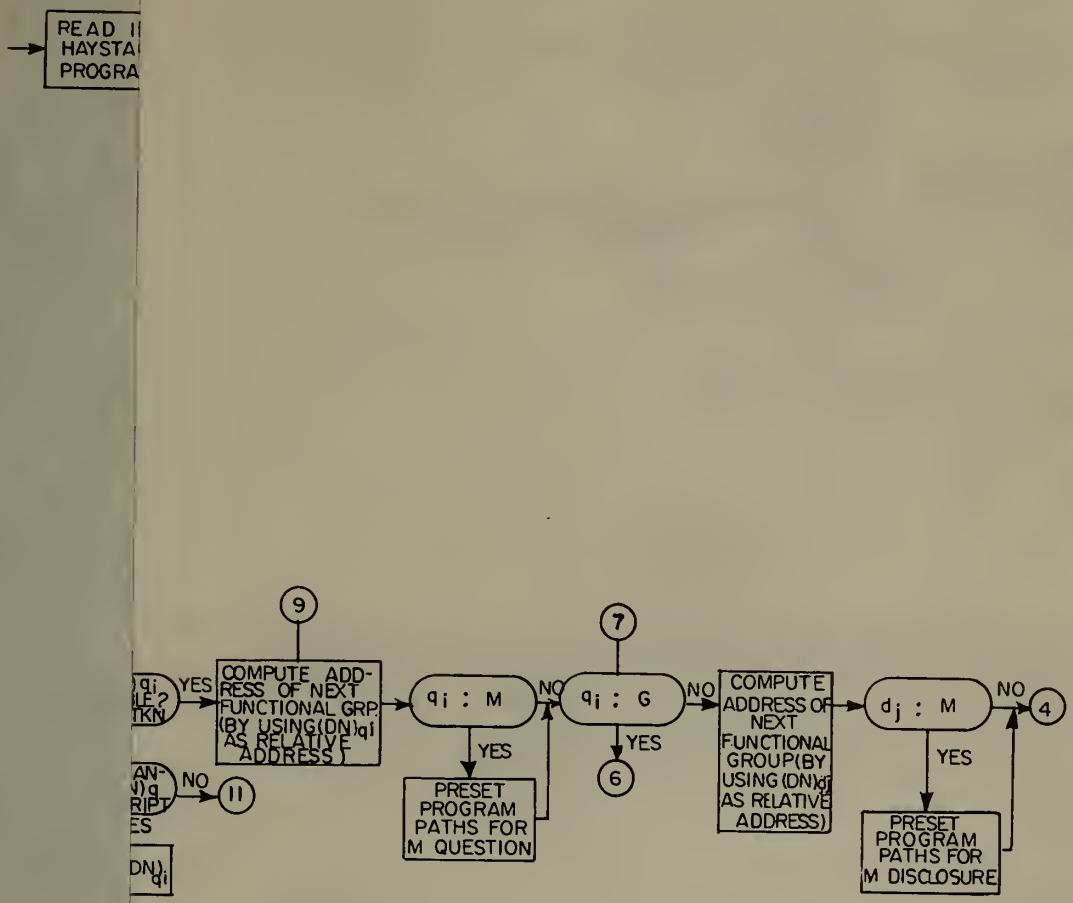
25. Robert T. Moore, "A Screening Method for Large Information Retrieval Systems", Proceedings of the Western Joint Computer Conference, Los Angeles, Calif., May 9-11, 1961.

26. E. Garfield, ed.-in-chief, "Encyclopedia Chimicus Internationalis (Cumulative Index Chemicus)", (Institute for Scientific Information, Philadelphia, Penna., 1962).

27. Austin M. Patterson, Leonard T. Capell, and Donald F. Walker, "The Ring Index", 2nd Ed., American Chemical Society, Washington, D.C., 1960.

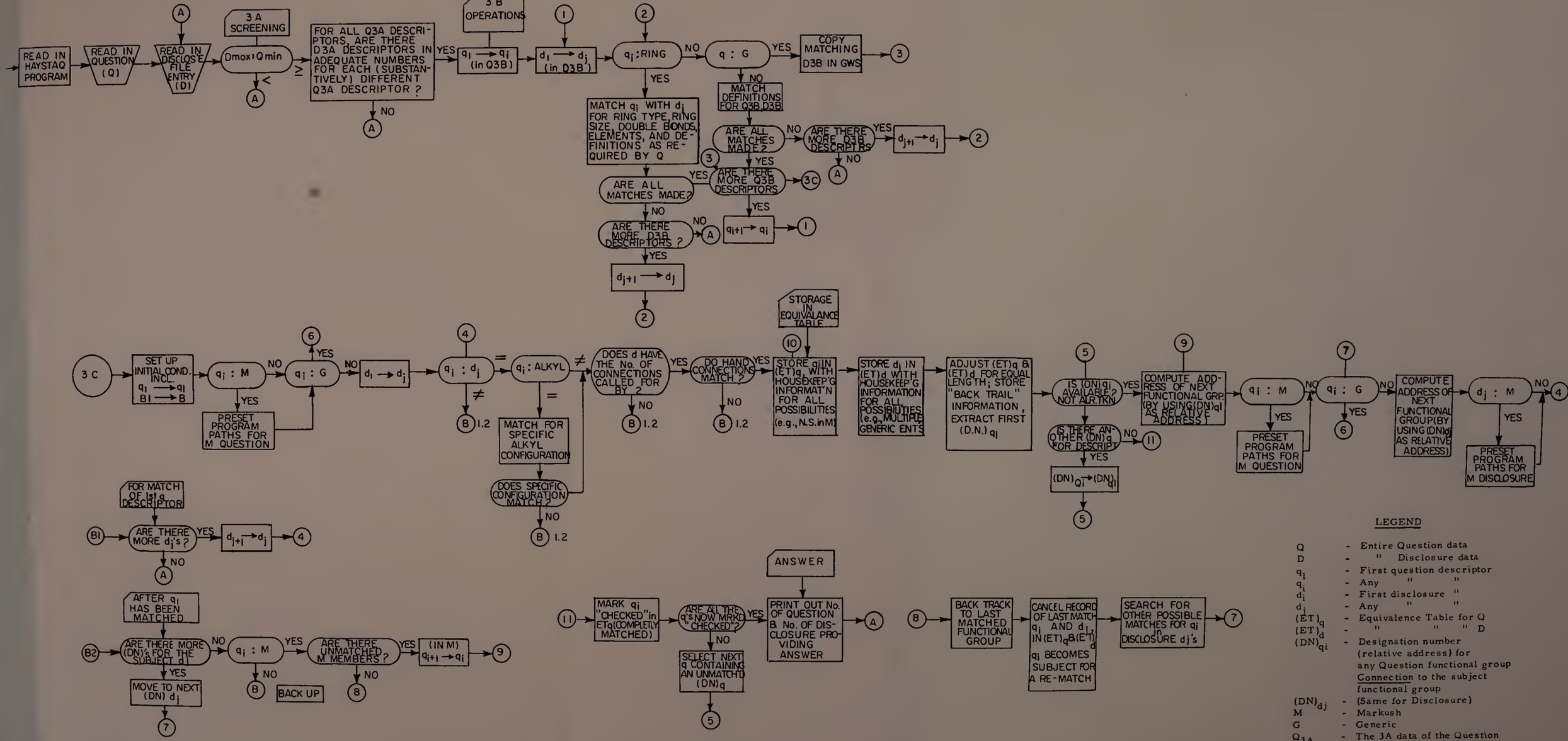
28. Commission on Codification, Ciphering, and Punched Card Techniques of the International Union of Pure and Applied Chemistry, "Rules for I.U.P.A.C. Notation for Organic Compounds", John Wiley and Sons, Inc., New York, N.Y., 1961.

☆ U.S. GOVERNMENT PRINTING OFFICE : 1965 O—786-795



LEGEND

Q	-	Entire Question data
D	-	" Disclosure data
q_1	-	First question descriptor
q_i	-	Any " "
d_i	-	First disclosure "
d_j	-	Any " "
$(ET)_q$	-	Equivalence Table for Q
$(ET)_d$	-	" " " D
$(DN)_{qi}$	-	Designation number (relative address) for any Question functional group Connection to the subject functional group
$(DN)_{dj}$	-	(Same for Disclosure)
M	-	Markush
G	-	Generic
Q_{3A}	-	The 3A data of the Question
D_{3A}	-	The 3A data of the Disclosure
Q_{3B}	-	The Question 3B Data
D_{3B}	-	The Disclosure 3B Data
GWS	-	Generic Word Storage
Q_{min}	-	Minimum # of functional groups required by Q
D_{max}	-	Maximum # of functional groups in D



LEGEND

Q	-	Entire Question data
D	-	" Disclosure data
q_1	-	First question descriptor
q_i	-	Any " "
d_1	-	First disclosure "
d_j	-	Any " "
$(ET)_q$	-	Equivalence Table for Q
$(ET)_d$	-	" " " D
$(DN)_{qi}$	-	Designation number (relative address) for any Question functional group in Connection to the subject functional group
$(DN)_{dj}$	-	(Same for Disclosure)
M	-	Markush
G	-	Generic
Q_{3A}	-	The 3A data of the Question
D_{3A}	-	The 3A data of the Disclosure
Q_{3B}	-	The Question 3B Data
D_{3B}	-	The Disclosure 3B Data
GWS	-	Generic Word Storage
Q_{min}	-	Minimum # of functional groups required by Q
D_{max}	-	Maximum # of functional groups in D

APPENDIX B

This flow chart is a greatly simplified version of the one employed as a working guide for HAYSTAQ. It is intended to illustrate the general logic of the algorithm employed, rather than to indicate in detail the complex interlocking relationships which the programmer must handle in writing machine instructions.

U.S. DEPARTMENT OF COMMERCE
WASHINGTON, D.C. 20230

OFFICIAL BUSINESS